Chapter 2
Hermite's Constant and Lattice Algorithms

Phong Q. Nguyen

Abstract We introduce lattices and survey the main provable algorithms for solving the shortest vector problem, either exactly or approximately. In doing so, we emphasize a surprising connection between lattice algorithms and the historical problem of bounding a well-known constant introduced by Hermite in 1850, which is related to sphere packings. For instance, we present the Lenstra–Lenstra–Lovász algorithm (LLL) as an (efficient) algorithmic version of Hermite's inequality on Hermite's constant. Similarly, we present blockwise generalizations of LLL as (more or less tight) algorithmic versions of Mordell's inequality.

Introduction

Informally, a *lattice* is an infinite arrangement of points in $\mathbb{R}^n$ spaced with sufficient regularity that one can shift any point onto any other point by some symmetry of the arrangement. The simplest example of a lattice is the hypercubic lattice $\mathbb{Z}^n$ formed by all points with integral coordinates. Geometry of numbers [1–4] is the branch of number theory dealing with lattices (and especially their connection with convex sets), and its origins go back to two historical problems:

1. *Higher-dimensional generalizations of Euclid's algorithm.* The elegance and simplicity of Euclid's greatest common divisor algorithm motivate the search for generalizations enjoying similar properties. By trying to generalize previous work of Fermat and Euler, Lagrange [5] studied numbers that can be represented by quadratic forms at the end of the eighteenth century: given a triplet $(a, b, c) \in \mathbb{Z}^3$, identify which integers are of the form $ax^2 + bxy + cy^2$, where $(x, y) \in \mathbb{Z}^2$. Fermat had for instance characterized numbers that are sums of two squares: $x^2 + y^2$, where $(x, y) \in \mathbb{Z}^2$. To answer such questions, Lagrange invented a generalization [5, pages 698–700] of Euclid's algorithm to binary quadratic forms. This algorithm is often attributed (incorrectly) to Gauss [6], and was generalized in

P.Q. Nguyen
INRIA, Ecole normale supérieure, Département d'informatique, 45 rue d'Ulm, 75005 Paris, France
e-mail: http://www.di.ens.fr/~pnguyen/

P.Q. Nguyen and B. Vallée (eds.), *The LLL Algorithm*, Information Security and Cryptography, DOI 10.1007/978-3-642-02295-1_2,

the nineteenth century by Hermite [7] to positive definite quadratic forms of arbitrary dimension. Let $q(x_1, \dots, x_n) = \sum_{1 \le i,j \le n} q_{i,j} x_i x_j$ be a positive definite quadratic form over $\mathbb{R}^n$, and denote by $\Delta(q) = \det_{1 \le i,j \le n} q_{i,j} \in \mathbb{R}^+$ its discriminant. Hermite [7] used his algorithm to prove that there exist $x_1, \dots, x_n \in \mathbb{Z}$ such that

$$0 < q(x_1, \dots, x_n) \le (4/3)^{(n-1)/2} \Delta(q)^{1/n}. \tag{2.1}$$

If we denote by $\|q\|$ the minimum of $q(x_1, \dots, x_n)$ over $\mathbb{Z}^n \setminus \{0\}$, (2.1) shows that $\|q\|/\Delta(q)^{1/n}$ can be upper bounded independently of q. This proves the existence of Hermite's constant γ_n defined as the supremum of this ratio over all positive definite quadratic forms:

$$\gamma_n = \max_{q \text{ positive definite over } \mathbb{R}^n} \frac{\|q\|}{\Delta(q)^{1/n}}, \tag{2.2}$$

because it turns out that the supremum is actually reached. The inequality (2.1) is equivalent to Hermite's inequality on Hermite's constant:

$$\gamma_n \le (4/3)^{(n-1)/2}, \ n \ge 1, \tag{2.3}$$

which can be rewritten as

$$\gamma_n \le \gamma_2^{n-1}, \ n \ge 1, \tag{2.4}$$

because Lagrange [5] showed that $\gamma_2 = \sqrt{4/3}$. Though Hermite's constant was historically defined in terms of positive definite quadratic forms, it can be defined equivalently using lattices, due to the classical connection between lattices and positive definite quadratic forms, which we will recall precisely in section "Quadratic Forms."

2. *Sphere packings.* This famous problem [8] asks what fraction of $\mathbb{R}^n$ can be covered by equal balls that do not intersect except along their boundaries. The problem is open as soon as $n \ge 4$ (see Fig. 2.1 for the densest packing for $n = 2$), which suggests to study simpler problems.

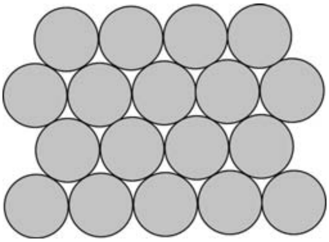

Fig. 2.1 The densest packing in dimension two: the hexagonal lattice packing

Of particular interest is the lattice packing problem, which asks what is the densest packing derived from lattices (such as the packing of Fig. 2.1): any full-rank lattice L induces a packing of $\mathbb{R}^n$ whose centers are the lattice points, and the diameter of the balls is the minimal distance $\lambda_1(L)$ between two lattice points. The density $\delta(L)$ of the lattice packing is equal to the ratio between the volume of the n-dimensional ball of diameter $\lambda_1(L)$ and the volume of any fundamental domain of L (i.e., the volume of the compact set $\mathbb{R}^n/L$). There is the following simple relationship between Hermite's constant γ_n and the supremum $\delta_n = \max_L \delta(L)$ over all full-rank lattices L of $\mathbb{R}^n$, due to the alternative lattice-based definition of γ_n previously mentioned:

$$\gamma_n = 4\left(\frac{\delta_n}{v_n}\right)^{2/n}, \tag{2.5}$$

where v_n denotes the volume of the n-dimensional unit ball. Thus, the problem of finding the maximal density of lattice packings is equivalent to finding the exact value of Hermite's constant γ_n, which is currently open for $n \geq 9, n \neq 24$.

Lattice algorithms deal with integral lattices, which are usually represented by a matrix with integer coefficients. This means that the lattice L is formed by all integral linear combinations of the row vectors of a given integral matrix B:

$$L = \{a_1\mathbf{b}_1 + \cdots + a_n\mathbf{b}_n, a_i \in \mathbb{Z}\},$$

where $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$ denote the row vectors of B. The most famous lattice problem is the so-called *shortest vector problem* (SVP), which asks to find a shortest nonzero vector in L, that is, a nonzero vector of the form $a_1\mathbf{b}_1 + \cdots + a_n\mathbf{b}_n$ (where $a_i \in \mathbb{Z}$) and of minimal Euclidean norm $\lambda_1(L)$. SVP can be viewed as a geometric generalization of gcd computations: Euclid's algorithm actually computes the smallest (in absolute value) nonzero linear combination of two integers, as $\gcd(a,b)\mathbb{Z} = a\mathbb{Z} + b\mathbb{Z}$, which means that we are replacing the integers a and b by an arbitrary number of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ with integer coordinates.

When the vectors $\mathbf{b}_i$'s span a low-dimensional space, one can solve SVP as efficiently as Euclid's algorithm. But when the dimension increases, NP-hardness looms (see [9]), which gives rise to two types of algorithms:

(a) *Exact algorithms.* These algorithms provably find a shortest vector, but they are expensive, with a running time at least exponential in the dimension. Intuitively, these algorithms perform an exhaustive search of all extremely short lattice vectors, whose number is exponential in the dimension (in the worst case): in fact, there are lattices for which the number of shortest lattice vectors is already exponential. The best deterministic algorithm is Kannan's enumeration [10,11], with super-exponential worst-case complexity, namely $n^{n/(2e)+o(n)}$ polynomial-time operations (see [12, 13]), where n denotes the lattice dimension. The best randomized algorithm is the sieve of Ajtai, Kumar, and Sivakumar (AKS) [14,15], with exponential worst-case complexity of $2^{O(n)}$ polynomial-time operations (where $O()$ can be taken to be 5.9 [15]): this algorithm also requires exponential space, whereas enumeration requires only negligible space.

(b) *Approximation algorithms*. The Lenstra–Lenstra–Lovász algorithm (LLL) and other efficient lattice reduction algorithms known provide only an approximation of SVP, in the sense that the norm of the nonzero output vector can be upper bounded using some function of the dimension, either absolutely or relatively to the minimal norm $\lambda_1(L)$. We will see that all polynomial-time approximation algorithms known [16–19] can be viewed as (more or less tight) algorithmic versions of upper bounds on Hermite's constant. For instance, LLL can be viewed as an algorithmic version of Hermite's inequality (2.3): it can be used to find efficiently $x_1, \ldots, x_n \in \mathbb{Z}$ satisfying essentially (2.1), which corresponds to short lattice vectors within Hermite's inequality. Similarly, the recent blockwise algorithm of Gama and Nguyen [19] can be viewed as an algorithmic version of Mordell's inequality, which itself is a generalization of Hermite's inequality (2.3).

In high dimension (say, higher than 150), only approximation algorithms are practical, but both categories are in fact complementary: all exact algorithms known first apply an approximation algorithm (typically at least LLL) as a preprocessing, while all approximation algorithms known call many times an exact algorithm in low dimension as a subroutine.

In this article, we will survey the main provable algorithms for solving the shortest vector problem, either exactly or approximately. This is related to Hermite's constant as follows:

- The analysis of exact algorithms involves counting the number of lattice points inside balls, for which good estimates are related to Hermite's constant.
- All approximation algorithms known are rather designed to find short nonzero lattice vectors in an absolute sense: the fact that the norm of the output is also relatively close to the first minimum can be viewed as a by-product. This means that any proof of correctness of the algorithm will have to include a proof that the output lattice vector is short in an absolute sense, which gives rise to an upper bound on Hermite's constant. In fact, it turns out that all approximation algorithms known are related (in a more or less tight manner) to a classical upper bound on Hermite's constant.

The rest of the article is organized as follows. Section "Background and Lattices" introduces lattices and their mathematical background. Section "Lattice Reduction" introduces lattice reduction and the main computational problems. Subsequent sections present the main lattice algorithms. Section "Two-Dimensional Case" deals with the two-dimensional case: Lagrange's algorithm. Section "Hermite's Inequality and the Lenstra–Lenstra–Lovász Algorithm" deals with the first efficient approximation algorithm in high dimension: the LLL algorithm. Section "Solving Exact SVP" deals with exact algorithms for SVP, which all use the LLL algorithm. Finally, section "Mordell's Inequality and Blockwise Algorithms" deals with polynomial-time generalizations of LLL that have a better approximation factor.

Background on Lattices

Notation

We consider $\mathbb{R}^n$ with its usual topology of an Euclidean vector space. We use bold letters to denote vectors, usually in row notation. The Euclidean inner product of two vectors $\mathbf{x} = (x_i)_{i=1}^n$ and $\mathbf{y} = (y_i)_{i=1}^n$ is denoted by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i .$$

The corresponding Euclidean norm is denoted by

$$\|\mathbf{x}\| = \sqrt{x_1^2 + \cdots + x_n^2}.$$

Denote by $\mathcal{B}(\mathbf{x}, r)$ the open ball of radius r centered at $\mathbf{x}$:

$$\mathcal{B}(\mathbf{x}, r) = \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{y}\| < r\}.$$

Definition 1. A subset D of $\mathbb{R}^n$ is called *discrete* when it has no limit point, that is, for all $x \in D$, there exists $\rho > 0$ such that $\mathcal{B}(x, \rho) \cap D = \{x\}$.

As an example, $\mathbb{Z}^n$ is discrete (because $\rho = 1/2$ clearly works), while $\mathbb{Q}^n$ and $\mathbb{R}^n$ are not. The set $\{1/n : n \in \mathbb{N}^*\}$ is discrete, but the set $\{0\} \cup \{1/n : n \in \mathbb{N}^*\}$ is not. Any subset of a discrete set is discrete.

For any ring R, we denote by $\mathcal{M}_{n,m}(R)$ (resp. $\mathcal{M}_n(R)$) the set of $n \times m$ (resp. $n \times n$) matrices with coefficients in R. $GL_n(R)$ denotes the group of invertible matrices in the ring $\mathcal{M}_n(R)$. For any subset S of $\mathbb{R}^n$, we define the linear span of S, denoted by $\mathrm{span}(S)$, as the minimal vector subspace (of $\mathbb{R}^n$) containing S.

Definition 2. Let $\mathbf{b}_1, \ldots, \mathbf{b}_m$ be in $\mathbb{R}^n$. The vectors $\mathbf{b}_i$'s are said to be *linearly dependent* if there exist $x_1, \ldots, x_m \in \mathbb{R}$, which are not all zero and such that

$$\sum_{i=1}^{m} x_i \mathbf{b}_i = 0.$$

Otherwise, they are said to be *linearly independent*.

Definition 3. The *Gram determinant* of $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{R}^n$, denoted by $\Delta(\mathbf{b}_1, \ldots, \mathbf{b}_m)$, is the determinant of the $m \times m$ Gram matrix $\left(\langle \mathbf{b}_i, \mathbf{b}_j \rangle\right)_{1 \le i,j \le m}$.

We list basic properties of the Gram determinant:

- The Gram determinant $\Delta(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ is always ≥ 0. It is equal to zero if and only if the $\mathbf{b}_i$'s are linearly dependent.

- The Gram determinant is invariant by any permutation of the m vectors, and by any integral linear transformation of determinant ± 1, such as adding to one of the vectors a linear combination of the others.
- The Gram determinant has a very important geometric interpretation: when the $\mathbf{b}_i$'s are linearly independent, $\sqrt{\Delta(\mathbf{b}_1, \ldots, \mathbf{b}_m)}$ is the m-dimensional volume $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ of the parallelepiped $\{\sum_{i=1}^{m} x_i \mathbf{b}_i : 0 \leq x_i \leq 1, 1 \leq i \leq m\}$ spanned by the $\mathbf{b}_i$'s.

Denote by v_n the volume of the n-dimensional unit ball $\mathcal{B}(0, 1)$. Then

$$v_n = \frac{\pi^{n/2}}{\Gamma(1 + n/2)} \sim \left(\frac{2e\pi}{n}\right)^{n/2} \frac{1}{\sqrt{\pi n}}, \tag{2.6}$$

where $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\,\mathrm{d}t$.

Lattices

Definition 4. A *lattice* of $\mathbb{R}^n$ is a discrete subgroup of $(\mathbb{R}^n, +)$; that is any subgroup of $(\mathbb{R}^n, +)$ which has the discreteness property.

Notice that an additive group is discrete if and only if 0 is not a limit point, which implies that a lattice is any nonempty set $L \subseteq \mathbb{R}^n$ stable by subtraction (in other words: for all $\mathbf{x}$ and $\mathbf{y}$ in L, $\mathbf{x}-\mathbf{y}$ belongs to L), and such that $L \cap \mathcal{B}(0, \rho) = \{0\}$ for some $\rho > 0$.

With this definition, the first examples of lattices that come to mind are the zero lattice $\{0\}$ and the *lattice of integers* $\mathbb{Z}^n$. Our definition implies that any subgroup of a lattice is a lattice, and therefore, any subgroup of $(\mathbb{Z}^n, +)$ is a lattice. Such lattices are called *integral lattices*. As an example, consider two integers a and $b \in \mathbb{Z}$: the set $a\mathbb{Z} + b\mathbb{Z}$ of all integral linear combinations of a and b is a subgroup of $\mathbb{Z}$, and therefore a lattice; it is actually the set $\gcd(a, b)\mathbb{Z}$ of all multiples of the gcd of a and b. For another example, consider n integers $a_1, \ldots, a_n$, together with a modulus M. Then the set of all $(x_1, \ldots, x_n) \in \mathbb{Z}^n$ such that $\sum_{i=1}^{n} a_i x_i \equiv 0 \pmod{M}$ is a lattice in $\mathbb{Z}^n$ because it is clearly a subgroup of $\mathbb{Z}^n$.

We give a few basic properties of lattices:

Lemma 1. *Let L be a lattice in $\mathbb{R}^n$.*

1. *There exists $\rho > 0$ such that for all $\mathbf{x} \in L$:*

$$L \cap \mathcal{B}(\mathbf{x}, \rho) = \{\mathbf{x}\}.$$

2. *L is closed.*
3. *For all bounded subsets S of $\mathbb{R}^n$, $L \cap S$ is finite.*
4. *L is countable.*

Notice that a set that satisfies either property 1 or 3 is necessarily discrete, but an arbitrary discrete subset of $\mathbb{R}^n$ does not necessarily satisfy property 1 nor 3. It is the group structure of lattices that allows such additional properties.

Bases

Let $\mathbf{b}_1, \ldots, \mathbf{b}_m$ be arbitrary vectors in $\mathbb{R}^n$. Denote by $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ the set of all integral linear combinations of the $\mathbf{b}_i$'s:

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m) = \left\{ \sum_{i=1}^{m} n_i \mathbf{b}_i : n_1, \ldots, n_m \in \mathbb{Z} \right\} . \tag{2.7}$$

This set is a subgroup of $\mathbb{R}^n$, but it is not necessarily discrete. For instance, one can show that $\mathcal{L}((1), (\sqrt{2}))$ is not discrete because $\sqrt{2} \notin \mathbb{Q}$. However, the following elementary result gives sufficient conditions for this set to be discrete:

Theorem 1. *The subgroup* $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ *is a lattice in either of the following two cases:*

1. $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{Q}^n$.
2. $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{R}^n$ *are linearly independent.*

Proof. Case 1 is trivial. Now consider Case 2, and let $L = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m)$. It suffices to show that 0 is not a limit point of L. Consider the parallelepiped P defined by

$$P = \left\{ \sum_{i=1}^{m} x_i \mathbf{b}_i : |x_i| < 1 \right\} .$$

As the $\mathbf{b}_i$'s are linearly independent, $L \cap P = \{0\}$. Besides, there exists $\rho > 0$ such that $B(0, \rho) \subseteq P$, which shows that 0 cannot be a limit point of L. □

Definition 5. When $L = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ is a lattice, we say that L is spanned by the $\mathbf{b}_i$'s, and that the $\mathbf{b}_i$'s are *generators*. When the $\mathbf{b}_i$'s are further linearly independent, we say that $(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ is a *basis* of the lattice L, in which case each lattice vector decomposes itself uniquely as an integral linear combination of the $\mathbf{b}_i$'s:

$$\forall \mathbf{v} \in L,\ \exists! v_1, \ldots, v_m \in \mathbb{Z} \text{ s.t. } \mathbf{v} = \sum_{i=1}^{m} v_i \mathbf{b}_i .$$

Bases and sets of generators are useful to represent lattices and to perform computations. One will typically represent a lattice on a computer by some lattice basis, which can itself be represented by a matrix with real coefficients. In practice, one will usually restrict to integral lattices, so that the underlying matrices are integral matrices.

Definition 6. We define the *dimension* or *rank* of a lattice L in $\mathbb{R}^n$, denoted by $\dim(L)$, as the dimension d of its linear span denoted by $\mathrm{span}(L)$. The lattice is said to be *full-rank* when $d = n$: in the remaining, we usually denote the dimension by n when the lattice is full-rank, and by d otherwise.

The dimension is the maximal number of linearly independent lattice vectors. Any lattice basis of L must have exactly d elements. There always exist d linearly independent lattice vectors; however, such vectors do not necessarily form a basis, as opposed to the case of vector spaces. But the following theorem shows that one can always derive a lattice basis from such vectors:

Theorem 2. *Let L be a d-dimensional lattice of $\mathbb{R}^n$. Let $\mathbf{c}_1, \ldots, \mathbf{c}_d \in L$ be linearly independent vectors. There exists a lower triangular matrix $(u_{i,j}) \in \mathcal{M}_d(\mathbb{R})$ such that the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$ defined as $\mathbf{b}_i = \sum_{j=1}^{i} u_{i,j}\mathbf{c}_j$ are linearly independent and such that $L = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_d)$.*

This proves the unconditional existence of lattice bases:

Corollary 1. *Any lattice of $\mathbb{R}^n$ has at least one basis.*

Thus, even if sets of the form $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ may or may not be lattices, all lattices can be written as $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ for some linearly independent $\mathbf{b}_i$'s. Corollary 1 together with Theorem 1 give an alternative definition of a lattice: a nonempty subset L of $\mathbb{R}^n$ is a lattice if only if there exist linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$ in $\mathbb{R}^n$ such that

$$L = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_d).$$

This characterization suggests that lattices are discrete analogues of vector spaces.

The following elementary result shows the relationship between two bases:

Theorem 3. *Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be a basis of a lattice L in $\mathbb{R}^n$. Let $\mathbf{c}_1, \ldots, \mathbf{c}_d$ be vectors of L. Then there exists a unique $d \times d$ integral matrix $U = (u_{i,j})_{1 \le i,j \le d} \in \mathcal{M}_d(\mathbb{Z})$ such that $\mathbf{c_i} = \sum_{j=1}^{d} u_{i,j}\mathbf{b}_j$ for all $1 \le i \le d$. And $(\mathbf{c}_1, \ldots, \mathbf{c}_d)$ is a basis of L if and only if the matrix U has determinant ± 1.*

As a result, as soon as the lattice dimension is ≥ 2, there are infinitely many lattice bases.

Quadratic Forms

Historically, lattices were first studied in the language of positive definite quadratic forms. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be a basis of a lattice L in $\mathbb{R}^n$. Then the function

$$q(x_1, \ldots, x_d) = \| \sum_{i=1}^{d} x_i \mathbf{b}_i \|^2, \tag{2.8}$$

defines a positive definite quadratic form over $\mathbb{R}^d$.

Reciprocally, let q be a positive definite quadratic form over $\mathbb{R}^d$. Then Cholesky factorization shows the existence of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$ of $\mathbb{R}^d$ such that (2.8) holds for all $(x_1, \ldots, x_d) \in \mathbb{R}^d$.

Volume and the Gaussian Heuristic

Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ and $(\mathbf{c}_1, \ldots, \mathbf{c}_d)$ be two bases of a lattice L in $\mathbb{R}^n$. By Theorem 3, there exists a $d \times d$ integral matrix $U = (u_{i,j})_{1 \leq i,j \leq d} \in \mathcal{M}_d(\mathbb{Z})$ of determinant ± 1 such that $\mathbf{c_i} = \sum_{j=1}^{d} u_{i,j} \mathbf{b}_j$ for all $1 \leq i \leq d$. It follows that the Gram determinant of those two bases are equal:

$$\Delta(\mathbf{b}_1, \ldots, \mathbf{b}_d) = \Delta(\mathbf{c}_1, \ldots, \mathbf{c}_d) > 0,$$

which gives rise to the following definition:

Definition 7. The *volume* (or *determinant*) of the lattice L is defined as

$$\mathrm{vol}(L) = \Delta(\mathbf{b}_1, \ldots, \mathbf{b}_d)^{1/2},$$

which is independent of the choice of the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of the lattice L.

We prefer the name *volume* to the name *determinant* because of its geometric interpretation: it corresponds to the d-dimensional volume of the parallelepiped spanned by any basis. In the mathematical literature, the lattice volume we have just defined is sometimes alternatively called co-volume, because it is also the volume of the torus $\mathrm{span}(L)/L$. For full-rank lattices, the volume has the following elementary properties:

Lemma 2. *Let L be a full-rank lattice in $\mathbb{R}^n$. Then:*

1. *For any basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L, $\mathrm{vol}(L) = |\det(\mathbf{b}_1, \ldots, \mathbf{b}_n)|$.*
2. *For any $r > 0$, denote by $s_L(r)$ the number of $\mathbf{x} \in L$ such that $\|\mathbf{x}\| \leq r$. Then*

$$\lim_{r \to \infty} \frac{s_L(r)}{r^n v_n} = 1/\mathrm{vol}(L).$$

The second statement of Lemma 2 says that, as the radius r grows to infinity, the number of lattice vectors inside the ball (centered at zero) of radius r is asymptotically equivalent to the ratio between the volume $r^n v_n$ of the n-dimensional ball of radius r and the volume of the lattice. This suggests the following heuristic, known as the Gaussian Heuristic:

Definition 8. Let L be a full-rank lattice in $\mathbb{R}^n$, and C be a measurable subset of $\mathbb{R}^n$. The *Gaussian Heuristic* "predicts" that the number of points of $L \cap C$ is roughly $\mathrm{vol}(C)/\mathrm{vol}(L)$.

We stress that this is only a heuristic: there are cases where the heuristic is proved to hold, but there are also cases where the heuristic is proved to be incorrect.

Given a lattice L, how does one compute the volume of L? If an explicit basis of L is known, this amounts to computing a determinant: for instance, the volume of the hypercubic lattice $\mathbb{Z}^n$ is clearly equal to one. But if no explicit basis is known, one can sometimes use full-rank sublattices, as we will see in the next subsection.

Sublattices

Definition 9. Let L be a lattice in $\mathbb{R}^n$. A *sublattice* of L is a lattice M included in L: clearly, the sublattices of L are the subgroups of L. If the rank of M is equal to the rank of L, we say that M is a *full-rank* sublattice of L.

Lemma 3. *Let L be a lattice in $\mathbb{R}^n$. A sublattice M of L is full-rank if and only if the group index $[L : M]$ is finite, in which case we have*

$$\mathrm{vol}(M) = \mathrm{vol}(L) \times [L : M].$$

As an illustration, consider n integers $a_1, \ldots, a_n$, together with a modulus M. We have seen in section "Lattices" that the set L of all $(x_1, \ldots, x_n) \in \mathbb{Z}^n$ such that $\sum_{i=1}^n a_i x_i \equiv 0 \pmod{M}$ is a lattice in $\mathbb{Z}^n$ because it is a subgroup of $\mathbb{Z}^n$. But there seems to be no trivial basis of L. However, note that $L \subseteq \mathbb{Z}^n$ and that the dimension of L is n because L contains all the vectors of the canonical basis of $\mathbb{R}^n$ multiplied by M. It follows that

$$\mathrm{vol}(L) = [\mathbb{Z}^n : L].$$

Furthermore, the definition of L clearly implies that

$$[\mathbb{Z}^n : L] = M / \gcd(M, a_1, a_2, \ldots, a_n).$$

Hence,

$$\mathrm{vol}(L) = \frac{M}{\gcd(M, a_1, a_2, \ldots, a_n)}.$$

Definition 10. A sublattice M of L is said to be *primitive* if there exists a subspace E of $\mathbb{R}^n$ such that $M = L \cap E$.

It follows from Theorem 2 that:

Lemma 4. *A sublattice M of L is primitive if and only if every basis of M can be completed to a basis of L, that is, for any basis $(\mathbf{b}_1, \ldots, \mathbf{b}_r)$ of M, there exist $\mathbf{b}_{r+1}, \ldots, \mathbf{b}_d \in L$ such that $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is a basis of L.*

Definition 11. Let $\mathbf{b}_1, \ldots, \mathbf{b}_k \in L$. They are *primitive vectors* of L if and only if $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is a primitive sublattice of L.

In particular, any nonzero shortest vector of L is primitive.

Projected Lattices

Let L be a lattice in $\mathbb{R}^n$. The (orthogonal) projection of L over a subspace of $\mathbb{R}^n$ is a subgroup of $\mathbb{R}^n$, but it is not necessarily discrete. However, with suitable choices of the subspace, one can ensure discreteness, in which case the projection is a lattice:

Lemma 5. *Let L be a d-rank lattice in $\mathbb{R}^n$, and M be a r-rank primitive sublattice of L: $1 \leq r \leq d$. Let π_M denote the orthogonal projection over the orthogonal supplement of the linear span of M. Then $\pi_M(L)$ is a lattice of $\mathbb{R}^n$, of rank $d - r$, and of volume* $\mathrm{vol}(L)/\mathrm{vol}(M)$.

Proof. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_r)$ be a basis of M. As M is primitive sublattice of L, this basis can be extended to a basis of L: there exist $\mathbf{b}_{r+1}, \ldots, \mathbf{b}_d \in L$ such that $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is a basis of L. Clearly, the set $\pi_M(L)$ is equal to $\mathcal{L}(\pi_M(\mathbf{b}_{r+1}), \ldots, \pi_M(\mathbf{b}_d))$. As $\pi_M(\mathbf{b}_{r+1}), \ldots, \pi_M(\mathbf{b}_{r+1})$ are linearly independent, the subgroup $\mathcal{L}(\pi_M(\mathbf{b}_{r+1}), \ldots, \pi_M(\mathbf{b}_d))$ is a lattice, and so is $\pi_M(L)$. □

The following corollary will be used many times in lattice reduction:

Corollary 2. *Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be a basis of a lattice L in $\mathbb{R}^n$. For $1 \leq i \leq d$, let π_i denote the orthogonal projection over the orthogonal supplement of the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$; in particular, π_1 is the identity. Then $\pi_i(L)$ is a lattice of $\mathbb{R}^n$, of rank $d - i + 1$, and of volume* $\mathrm{vol}(L)/\mathrm{vol}(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}))$.

We will often use the notation π_i.

It is classical to prove statements by induction on the lattice rank using projected lattices, such as in the classical proof of Hermite's inequality: see Theorem 8 of section "Hermite's Inequality and the Lenstra–Lenstra–Lovász Algorithm." More precisely, for any basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L, we have $\dim(\pi_2(L)) = \dim(L) - 1$, and any nonzero vector $\mathbf{v} \in \pi_2(L)$ can be lifted into a nonzero vector $\mathbf{u} \in L$ such that $\mathbf{v} = \pi_2(\mathbf{u})$ and $\|\mathbf{u}\|^2 \leq \|\mathbf{v}\|^2 + \|\mathbf{b}_1\|^2/4$. This means that if one can find a short vector in $\pi_2(L)$, then one can also find a reasonably short vector in L.

Duality

Let L be a lattice in $\mathbb{R}^n$. The *dual lattice* of L is defined as

$$L^\times = \{\mathbf{y} \in \mathrm{span}(L) \text{ such that} \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z} \text{ for all } \mathbf{x} \in L\}.$$

Lemma 6. *If L is a d-rank lattice of $\mathbb{R}^n$, then $L^\times$ is a d-rank lattice of $\mathbb{R}^n$ such that*

$$\mathrm{vol}(L) \times \mathrm{vol}(L^\times) = 1.$$

Duality also allows to consider sublattices of lower dimension, which can be used in proofs by induction, such as the classical proof of Mordell's inequality (see section "Classical Proofs of Mordell's Inequality"). For instance, if L is a d-rank lattice and $\mathbf{v}$ is a nonzero vector of $L^\times$, then $L \cap H$ is a $(d-1)$-rank sublattice of L, where $H = \mathbf{v}^\perp$ denotes the hyperplane orthogonal to $\mathbf{v}$.

Gram–Schmidt and Triangularization

Definition 12. Let $\mathbf{b}_1, \ldots, \mathbf{b}_d$ be linearly independent vectors in $\mathbb{R}^n$. Their *Gram–Schmidt orthogonalization* (GSO) is the orthogonal family $(\mathbf{b}_1^\star, \ldots, \mathbf{b}_d^\star)$ defined as follows: $\mathbf{b}_1^\star = \mathbf{b}_1$ and more generally $\mathbf{b}_i^\star = \pi_i(\mathbf{b}_i)$ for $1 \le i \le d$, where π_i denotes (as in Corollary 2) the orthogonal projection over the orthogonal supplement of the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$.

We have the recursive formula

$$\mathbf{b}_i^\star = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^\star, \quad \text{where } \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^\star \rangle}{\left\| \mathbf{b}_j^\star \right\|^2} \text{ for all } 1 \le j < i \le d \tag{2.9}$$

The main reason why the Gram–Schmidt orthogonalization is widely used in lattice reduction is because it allows to triangularize the basis. More precisely, the family $(\mathbf{b}_1^\star/\|\mathbf{b}_1^\star\|, \ldots, \mathbf{b}_d^\star/\|\mathbf{b}_d^\star\|)$ is an orthonormal basis of $\mathbb{R}^n$. And if we express the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$ with respect to the orthonormal basis $(\mathbf{b}_1^\star/\|\mathbf{b}_1^\star\|, \ldots, \mathbf{b}_d^\star/\|\mathbf{b}_d^\star\|)$ (rather than the canonical basis), we obtain the following lower-triangular matrix, with diagonal coefficients $\|\mathbf{b}_1^\star\|, \ldots, \|\mathbf{b}_d^\star\|$:

$$\begin{pmatrix} \|\mathbf{b}_1^\star\| & 0 & \ldots & 0 \\ \mu_{2,1}\|\mathbf{b}_1^\star\| & \|\mathbf{b}_2^\star\| & \ddots & \\ & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mu_{d,1}\|\mathbf{b}_1^\star\| & \ldots & \mu_{d,d-1}\|\mathbf{b}_{d-1}^\star\| & \|\mathbf{b}_d^\star\| \end{pmatrix} \tag{2.10}$$

This can be summarized by the matrix equality $B = \mu B^\star$, where B is the $d \times n$ matrix whose rows are $\mathbf{b}_1, \ldots, \mathbf{b}_d$, $B^\star$ is the $d \times n$ matrix whose rows are $\mathbf{b}_1^\star, \ldots, \mathbf{b}_d^\star$, and μ is the $d \times d$ lower-triangular matrix whose diagonal coefficients are all equal to 1, and whose off-diagonal coefficients are the $\mu_{i,j}$'s. It follows that the lattice L spanned by the $\mathbf{b}_i$'s satisfies

$$\mathrm{vol}(L) = \prod_{i=1}^{d} \|\mathbf{b}_i^\star\| \tag{2.11}$$

Notice that the GSO family depends on the order of the vectors. If $\mathbf{b}_i \in \mathbb{Q}^n$, then $\mathbf{b}_i^\star \in \mathbb{Q}^n$ and $\mu_{i,j} \in \mathbb{Q}$. The GSO of $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is $(\mu_{i,j})_{1 \le j < i \le d}$ together with $(\|\mathbf{b}_i^\star\|)_{1 \le i \le d}$. Geometrically, $\|\mathbf{b}_i^\star\|$ is the distance of $\mathbf{b}_i$ to span$\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$.

The basis triangularization could have been obtained with other factorizations. For instance, if we had used Iwasa's decomposition of the row matrix B corresponding to $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, we would have obtained $B = UDO$, where U is a lower-triangular matrix with unit diagonal, D is diagonal, and O is an orthogonal matrix. In other words, U would be the matrix defined by the $\mu_{i,j}$'s (lower-triangular with unit diagonal, where the remaining coefficients are the $\mu_{i,j}$'s), D would be the diagonal matrix defined by the $\|\mathbf{b}_i^\star\|$'s, and O would be the row representation of $(\mathbf{b}_1^\star/\|\mathbf{b}_1^\star\|, \ldots, \mathbf{b}_d^\star/\|\mathbf{b}_d^\star\|)$.

Finally, it is worth noting that Gram–Schmidt orthogonalization is related to duality as follows. For any $i \in \{2, \ldots, d\}$, the vector $\mathbf{b}_i^\star/\|\mathbf{b}_i^\star\|^2$ is orthogonal to $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$ and we have $\langle \mathbf{b}_i^\star/\|\mathbf{b}_i^\star\|^2, \mathbf{b}_i \rangle = 1$, which implies that

$$\mathbf{b}_i^\star/\|\mathbf{b}_i^\star\|^2 \in \pi_j(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i))^\times, \forall j \in \{1, \ldots, i\}.$$

Lattice Reduction

A fundamental result of linear algebra states that any finite-dimensional vector space has a basis. We earlier established the analogue result for lattices: any lattice has a basis. In the same vein, a fundamental result of bilinear algebra states that any finite-dimensional Euclidean space has an orthonormal basis, that is, a basis consisting of unit vectors that are pairwise orthogonal. A natural question is to ask whether lattices also have orthonormal bases, or at least, orthogonal bases. Unfortunately, it is not difficult to see that even in dimension two, a lattice may not have an orthogonal basis, and this is in fact a typical situation. Informally, the goal of lattice reduction is to circumvent this problem: more precisely, the theory of lattice reduction shows that in any lattice, there is always a basis, which is not that far from being orthogonal. Defining precisely what is meant exactly by not being far from being orthogonal is tricky, so for now, let us just say that such a basis should consist of reasonably short lattice vectors, which implies that geometrically such vectors are not far from being orthogonal to each other.

Minkowski's Minima

To explain what is a reduced basis, we need to define what is meant by short lattice vectors. Let L be a lattice of dimension ≥ 1 in $\mathbb{R}^n$. There exists a nonzero vector $\mathbf{u} \in L$. Consider the closed hyperball $\mathcal{B}$ of radius $\|\mathbf{u}\|$ centered at zero. By Lemma

1, $L \cap \mathcal{B}$ is finite and contains $\mathbf{u}$, so it must have a shortest nonzero vector. The Euclidean norm of that shortest nonzero vector is called the *first minimum* of L, and is denoted by $\lambda_1(L) > 0$ or $\|L\|$. By definition, any nonzero vector $\mathbf{v}$ of L satisfies $\|\mathbf{v}\| \geq \lambda_1(L)$, and $\lambda_1(L)$ is the minimal distance between two distinct lattice vectors. And there exists $\mathbf{w} \in L$ such that $\|\mathbf{w}\| = \lambda_1(L)$: any such $\mathbf{w}$ is called a shortest vector of L, and it is not unique as $-\mathbf{w}$ would also be a shortest vector. The *kissing number* of L is the number of shortest vectors in L: it is upper bounded by some exponential function of the lattice dimension (see [8]).

We noticed that if $\mathbf{w}$ is a shortest vector of L, then so is $-\mathbf{w}$. Thus, one must be careful when defining the *second-to-shortest* vector of a lattice. To circumvent this problem, Minkowski [1] defined the other minima as follows.

Definition 13. Let L be a lattice of $\mathbb{R}^n$. For all $1 \leq i \leq \dim(L)$, the ith *minimum* $\lambda_i(L)$ is defined as the minimum of $\max_{1 \leq j \leq i} \|\mathbf{v}_j\|$ over all i linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_i \in L$.

Clearly, the minima are increasing: $\lambda_1(L) \leq \lambda_2(L) \leq \cdots \leq \lambda_d(L)$. And the Gram–Schmidt triangularization implies:

Lemma 7. *If $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is a basis of a lattice L, then its GSO satisfies for all $1 \leq i \leq d$*

$$\lambda_i(L) \geq \min_{i \leq j \leq d} \|\mathbf{b}_j^\star\|.$$

It is not difficult to see that there always exist linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ reaching simultaneously the minima, that is, $\|\mathbf{v}_i\| = \lambda_i(L)$ for all i. However, surprisingly, as soon as $\dim(L) \geq 4$, such vectors do not necessarily form a lattice basis. The canonical example is the four-dimensional lattice L defined as the set of all $(x_1, x_2, x_3, x_4) \in \mathbb{Z}^4$ such that $\sum_{i=1}^4 x_i$ is even. It is not difficult to see that $\dim(L) = 4$ and that all the minima of L are equal to $\sqrt{2}$. Furthermore, it can be checked that the following row vectors form a basis of L:

$$\begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

The basis proves in particular that $\mathrm{vol}(L) = 2$. However, the following row vectors are linearly independent lattice vectors, which also reach all the minima:

$$\begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}.$$

But they do not form a basis, as their determinant is equal to 4: another reason is that for all such vectors, the sum of the first two coordinates is even, and that property also holds for any integral linear combination of those vectors, but clearly not for all vectors of the lattice L. More precisely, the sublattice spanned by those four row vectors has index two in the lattice L.

Nevertheless, in the lattice L, there still exists at least one basis that reaches all the minima simultaneously, and we already gave one such basis. This also holds for any lattice of dimension ≤ 4, but it is no longer true in dimension ≥ 5, as was first noticed by Korkine and Zolotarev in the nineteenth century, in the language of quadratic forms. More precisely, it can easily be checked that the lattice spanned by the rows of the following matrix

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

has no basis reaching all the minima (which are all equal to two).

Hermite's Constant and Minkowski's Theorems

Now that successive minima have been defined, it is natural to ask how large those minima can be. Hermite [7] was the first to prove that the quantity $\lambda_1(L)/\mathrm{vol}(L)^{1/d}$ could be upper bounded over all d-rank lattices L.

Definition 14. The supremum of $\lambda_1(L)^2/\mathrm{vol}(L)^{2/d}$ over all d-rank lattices L is denoted by γ_d, and called *Hermite's constant* of dimension d.

The use of quadratic forms in [7] explains why Hermite's constant refers to $\max_L \lambda_1(L)^2/\mathrm{vol}(L)^{2/d}$ and not to $\max_L \lambda_1(L)/\mathrm{vol}(L)^{1/d}$. It can be noted that γ_d could also be equivalently defined as the supremum of $\lambda_1(L)^2$ over all d-rank lattices L of unit volume.

It is known that γ_d is reached, that is, for all $d \geq 1$, there is a d-rank lattice L such that $\gamma_d = \lambda_1(L)^2/\mathrm{vol}(L)^{2/d}$, and any such lattice is called *critical*. But finding the exact value of γ_d is a very difficult problem, which has been central in Minkowski's geometry of numbers. The exact value of γ_d is known only for $1 \leq d \leq 8$ (see the book [20] for proofs) and very recently also for $d = 24$ (see [21]): the values are summarized in the following table.

d	2	3	4	5	6	7	8	24
γ_d	$2/\sqrt{3}$	$2^{1/3}$	$\sqrt{2}$	$8^{1/5}$	$(64/3)^{1/6}$	$64^{1/7}$	2	4
Approximation	1.1547	1.2599	1.4142	1.5157	1.6654	1.8114	2	4

Furthermore, the list of all critical lattices (up to scaling and isometry) is known for each of those dimensions.

However, rather tight asymptotical bounds are known for Hermite's constant. More precisely, we have

$$\frac{d}{2\pi e}+\frac{\log(\pi d)}{2\pi e}+o(1)\leq\gamma_d\leq\frac{1.744d}{2\pi e}(1+o(1)).$$

For more information on the proof of those bounds: see [22, Chap. 2] for the lower bound (which comes from the Minkowski–Hlawka theorem), and [8, Chap. 9] for the upper bound. Thus, γ_d is essentially linear in d. It is known that $\gamma_d^d\in\mathbb{Q}$ (because there is always an integral critical lattice), but it is unknown if $(\gamma_d)_{d\geq 1}$ is an increasing sequence.

Hermite's historical upper bound [7] on his constant was exponential in the dimension:

$$\gamma_d\leq(4/3)^{(d-1)/2}.$$

The first linear upper bound on Hermite's constant is due to Minkowski, who viewed it as a consequence of his Convex Body Theorem:

Theorem 4 (Minkowski's Convex Body Theorem). *Let L be a full-rank lattice of $\mathbb{R}^n$. Let C be a measurable subset of $\mathbb{R}^n$, convex, symmetric with respect to 0, and of measure $> 2^n\mathrm{vol}(L)$. Then C contains at least a nonzero point of L.*

This theorem is a direct application of the following elementary lemma (see [2]), which can be viewed as a generalization of the pigeon-hole principle:

Lemma 8 (Blichfeldt). *Let L be a full-rank lattice in $\mathbb{R}^n$, and F be a measurable subset of $\mathbb{R}^n$ with measure $> \mathrm{vol}(L)$. Then F contains at least two distinct vectors whose difference belongs to L.*

Indeed, we may consider $F=\frac{1}{2}C$, and the assumption in Theorem 4 implies that the measure of F is $> \mathrm{vol}(L)$. From Blichfeldt's lemma, it follows that there exist $\mathbf{x}$ and $\mathbf{y}$ in F such that $\mathbf{x}-\mathbf{y}\in L\setminus\{0\}$. But

$$\mathbf{x}-\mathbf{y}=\frac{1}{2}(2\mathbf{x}-2\mathbf{y}),$$

which belongs to C by convexity and symmetry with respect to 0. Hence, $\mathbf{x}-\mathbf{y}\in C\cap(L\setminus\{0\})$, which completes the proof of Theorem 4.

One notices that the bound on the volumes in Theorem 4 is the best possible, by considering

$$C=\left\{\sum_{i=1}^{n}x_i\mathbf{b}_i\ :|x_i|<1\right\},$$

where the $\mathbf{b}_i$'s form an arbitrary basis of the lattice. Indeed, the measure of this C is exactly $2^n\mathrm{vol}(L)$, but by definition of C, no nonzero vector of L belongs to C.

In Theorem 4, the condition on the measure of C is a strict inequality, but it is not difficult to show that the strict inequality can be relaxed to an inequality $\geq 2^n\mathrm{vol}(L)$ if C is further assumed to be compact. By choosing for C a closed hyperball of sufficiently large radius (so that the volume inequality is satisfied), one obtains the following corollary:

Corollary 3. *Any d-dimensional lattice L of $\mathbb{R}^n$ contains a nonzero $\mathbf{x}$ such that*

$$\|\mathbf{x}\| \leq 2 \left(\frac{\mathrm{vol}(L)}{v_d} \right)^{\frac{1}{d}},$$

where v_d denotes the volume of the closed unitary hyperball of $\mathbb{R}^d$. In other words,

$$\gamma_d \leq \left(\frac{4}{v_d} \right)^{2/d}, \quad d \geq 1.$$

Note that if the Gaussian heuristic (see Definition 8 of section "Volume and the Gaussian Heuristic") held for all hyperballs, we would expect $\lambda_1(L)$ to be close to $(\mathrm{vol}(L)/v_d)^{1/d} \approx \sqrt{d/(2\pi e)}\mathrm{vol}(L)^{1/d}$ by (2.6). This means that the proved upper bound is only twice as large as the heuristic estimate from the Gaussian heuristic.

Using well-known formulas for v_d, one can derive a linear bound on Hermite's constant, for instance

$$\forall d, \; \gamma_d \leq 1 + \frac{d}{4}.$$

Notice that this bound is reached by $L = \mathbb{Z}^d$.

Now that we know how to bound the first minimum, it is natural to ask if a similar bound can be obtained for the other minima. Unfortunately, one cannot hope to upper bound separately the other minima, because the successive minima could be unbalanced. For instance, consider the rectangular two-rank lattice L spanned by the following row matrix:

$$\begin{pmatrix} \varepsilon & 0 \\ 0 & 1/\varepsilon \end{pmatrix},$$

where $\varepsilon > 0$ is small. The volume of L is one, and by definition of L, it is clear that $\lambda_1(L) = \varepsilon$ and $\lambda_2(L) = 1/\varepsilon$ if $\varepsilon \leq 1$. Here, $\lambda_2(L)$ can be arbitrarily large compared to the lattice volume, while $\lambda_1(L)$ can be arbitrarily small compared to the upper bound given by Hermite's constant.

However, it is always possible to upper bound the geometric mean of the first consecutive minima, as summarized by the following theorem (for an elementary proof, see [2**?**]):

Theorem 5 (Minkowski's Second Theorem). *Let L be a d-rank lattice of $\mathbb{R}^n$. Then for any integer r such that $1 \leq r \leq d$,*

$$\left(\prod_{i=1}^{r} \lambda_i(L) \right)^{1/r} \leq \sqrt{\gamma_d}\mathrm{vol}(L)^{1/d}.$$

Rankin's Constant

In 1953, Rankin [24] introduced the following generalization of Hermite's constant. For any n-rank lattice L and $1 \leq m \leq n$, the Rankin invariant $\gamma_{n,m}(L)$ is defined as

$$\gamma_{n,m}(L) = \min_{\substack{\mathbf{x}_1,\ldots,\mathbf{x}_m \in L \\ \mathrm{vol}(\mathbf{x}_1,\ldots,\mathbf{x}_m) \neq 0}} \left(\frac{\mathrm{vol}(\mathbf{x}_1,\ldots,\mathbf{x}_m)}{\mathrm{vol}(L)^{m/n}}\right)^2$$

$$= \min_{\substack{S \text{ sublattice of } L \\ \dim S = m}} \left(\frac{\mathrm{vol}(M)}{\mathrm{vol}(L)^{m/n}}\right)^2 \tag{2.12}$$

Using a family of linearly independent lattice vectors simultaneously reaching all the minima and Theorem 5, one obtains

$$\gamma_{n,m}(L) \leq \left(\frac{\prod_{i=1}^{m} \lambda_i(L)}{\mathrm{vol}(L)^{m/n}}\right)^2 \leq \gamma_n^m.$$

It follows that Rankin's constant $\gamma_{n,m} = \max \gamma_{n,m}(L)$ over all n-rank lattices L is well-defined, and we have $\gamma_{n,m} \leq \gamma_n^m$. This upper bound is not tight: using HKZ reduction (which we will define later) as in [17, 18], it can be shown that for $1 \leq m \leq n/2$,

$$\gamma_{n,m} \leq O(n)^{(n-m)\times(1/(n-1)+1/(n-2)+\cdots+1/(n-m))} \tag{2.13}$$

Rankin's constants satisfy the following three relations, which are proved in [20, 24]:

$$\forall n \in \mathbb{N},\ \gamma_{n,n} = 1,\ \gamma_{n,1} = \gamma_n \tag{2.14}$$

$$\forall n, m \text{ with } m < n\ \gamma_{n,m} = \gamma_{n,n-m} \tag{2.15}$$

$$\forall r \in [m+1, n-1],\ \gamma_{n,m} \leq \gamma_{r,m}(\gamma_{n,r})^{m/r} \tag{2.16}$$

The only known values of Rankin's constants are $\gamma_{4,2} = \frac{3}{2}$, which is reached for the $\mathbb{D}_4$ lattice, and those corresponding to the nine Hermite constants known. In the definition of $\gamma_{n,m}(L)$, the minimum is taken over sets of m linearly independent vectors of L, but we may restrict the definition to primitive sets of L or pure sublattices of L, as for any sublattice S of L, there exists a pure sublattice S_1 of L with $\mathrm{span}(S) = \mathrm{span}(S_1)$ and $\mathrm{vol}(S)/\mathrm{vol}(S_1) = [S : S_1]$. If $\mathrm{vol}(S)$ is minimal, then $[S : S_1] = 1$ so $S = S_1$ is pure.

Thunder [25] and Bogulavsky [26] proved the following lower bound on Rankin's constant, as a generalization of Minkowski–Hlawka's theorem:

$$\gamma_{n,m} \geq \left(n \frac{\prod_{j=n-m+1}^{n} Z(j)}{\prod_{j=2}^{m} Z(j)} \right)^{\frac{2}{n}} \tag{2.17}$$

where $Z(j) = \zeta(j)\Gamma(\frac{j}{2})/\pi^{\frac{j}{2}}$ and ζ is Riemann's zeta function: $\zeta(j) = \sum_{p=1}^{\infty} p^{-j}$. This shows that for $1 \leq m \leq n/2$,

$$\gamma_{n,m} \geq \Omega(n)^{m(n-m+1)/n} \tag{2.18}$$

Hermite–Korkine–Zolotarev (HKZ) Reduction

Hermite [7] introduced the following weak reduction notion in the language of quadratic forms:

Definition 15. A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice is *size-reduced* if its Gram–Schmidt orthogonalization satisfies, for all $1 \leq j < i \leq d$,

$$|\mu_{i,j}| \leq \frac{1}{2}. \tag{2.19}$$

Geometrically, this means that the projection $\mathbf{b}_i - \mathbf{b}_i^\star$ of $\mathbf{b}_i$ over the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$ is inside the parallelepiped $\mathcal{P} = \{\sum_{j=1}^{i-1} x_i \mathbf{b}_i,\ |x_j| \leq 1/2\}$ spanned by $\mathbf{b}_1^\star, \ldots, \mathbf{b}_{i-1}^\star$ with coefficients $\leq 1/2$ in absolute value, one tries to reduce the component of $\mathbf{b}_i$ over the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. Then (2.19) implies for all $1 \leq i \leq d$:

$$\|\mathbf{b}_i^\star\|^2 \leq \|\mathbf{b}_i\|^2 \leq \|\mathbf{b}_i^\star\|^2 + \frac{1}{4} \sum_{j=1}^{i-1} \|\mathbf{b}_j^\star\|^2. \tag{2.20}$$

Korkine and Zolotarev [27,28] strengthened Hermite's size-reduction as follows:

Definition 16. A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice is *Hermite–Korkine–Zolotarev-reduced (HKZ-reduced)* if it is size-reduced and such that for all $1 \leq i \leq d$, $\|\mathbf{b}_i^\star\| = \lambda_1(\pi_i(L))$.

Note that $\mathbf{b}_i^\star \in \pi_i(L)$ and $\mathbf{b}_i^\star \neq 0$, so it is natural to ask that $\|\mathbf{b}_i^\star\| = \lambda_1(\pi_i(L))$. Note also that the condition $\|\mathbf{b}_d^\star\| = \lambda_1(\pi_d(L))$ is necessarily satisfied.

HKZ-reduced bases have two interesting properties. The first is that an HKZ-reduced basis provides a very good approximation to the successive minima:

Theorem 6. *Let* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ *be an HKZ-reduced basis of a lattice* L*, then for all index* i *such that* $1 \leq i \leq d$,

$$\frac{4}{i+3} \leq \left(\frac{\|\mathbf{b}_i\|}{\lambda_i(L)} \right)^2 \leq \frac{i+3}{4}$$

The upper bound is easy to prove and can be attributed to Mahler [29]: it suffices to notice that $\|\mathbf{b}_i^\star\| = \lambda_1(\pi_i(L)) \leq \lambda_i(L)$ (where the right-hand inequality can be proved by considering a set of linearly independent vectors reaching all the minima simultaneously), and to use the right-hand inequality of (2.20). The lower bound is proved in [30]: first, notice that HKZ-reduction implies that for all $1 \leq j \leq i$, $\|\mathbf{b}_j^\star\| \leq \|\mathbf{b}_i\|$, therefore $\|\mathbf{b}_j\|^2/\|\mathbf{b}_i\|^2 \leq (j+3)/4$ by (2.20). It should be noted that it is not necessarily true that $\|\mathbf{b}_i\| \geq \lambda_i(L)$ because it does not necessarily hold that $\|\mathbf{b}_2\| \leq \|\mathbf{b}_3\| \leq \cdots \leq \|\mathbf{b}_d\|$. Thus, the gap between an HKZ-reduced basis and the successive minima of a lattice is at most polynomial, namely less than $\sqrt{(i+3)/4}$. The article [30] shows that the bounds of Theorem 6 are not far from being tight in the worst case.

The second interesting property of HKZ-reduced bases is that they have local properties. Indeed, if $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is HKZ-reduced, then $(\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \ldots, \pi_i(\mathbf{b}_j))$ is HKZ-reduced for all $1 \leq i \leq j \leq d$. Thus, by studying low-dimensional HKZ-reduced bases, one can deduce properties holding for any dimension. For instance, any two-dimensional HKZ-reduced basis $(\mathbf{c}_1, \mathbf{c}_2)$ satisfies $\|\mathbf{c}_1\|/\|\mathbf{c}_2^\star\| \leq \sqrt{4/3}$, which implies that ay HKZ-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ satisfies $\|\mathbf{b}_i^\star\|/\|\mathbf{b}_{i+1}^\star\| \leq \sqrt{4/3}$ for all $1 \leq i \leq d$. It is by using such reasonings that Korkine and Zolotarev found better upper bounds on Hermite's constant than Hermite's inequality.

Algorithmic Lattice Problems

In the previous section, we presented lattice reduction from a mathematical point of view. In this section, we introduce the main algorithmic problems for lattices.

Representation

In practice, one deals only with *rational lattices*, that is, lattices included in $\mathbb{Q}^n$. In this case, by a suitable multiplication, one needs only to be able to deal with integral lattices, those which are included in $\mathbb{Z}^n$. Such lattices are usually represented by a basis, that is, a matrix with integral coefficients. When we explicitly give such a matrix, we will adopt a row representation: the row vectors of the matrix will be the basis vectors. The size of the lattice is measured by the dimensions of the matrix (the number d of rows, which correspond to the lattice dimension, and the number n of columns), and the maximal bit-length $\log B$ of the matrix coefficients; thus, the whole matrix can be stored using $dn \log B$ bits.

Lattice problems are often relative to norms: here, we will only be concerned with the Euclidean norm. Before describing hard problems, let us recall two easy problems that can be solved in deterministic polynomial time:

- Given a generating set of an integral lattice L, find a basis of the lattice L.
- Given a basis of an integral lattice $L \subseteq \mathbb{Z}^n$ and a target vector $\mathbf{v} \in \mathbb{Z}^n$, decide if $\mathbf{v} \in L$, and if so, find the decomposition of $\mathbf{v}$ with respect to the basis.

The Shortest Vector Problem (SVP)

The most famous lattice problem is the following:

Problem 1 (Shortest Vector Problem (SVP)). Given a basis of a d-rank integral lattice L, find $\mathbf{u} \in L$ such that $\|\mathbf{u}\| = \lambda_1(L)$.

In its exact form, this problem is known to be NP-hard under randomized reductions (see the survey [9]), which suggests to relax the problem. There are two approximation versions of SVP: approx-SVP (ASVP) and Hermite-SVP (HSVP), which are defined below.

Problem 2 (Approximate Shortest Vector Problem (ASVP)). Given a basis of a d-rank integral lattice L and an approximation factor $f \geq 1$, find a nonzero $\mathbf{u} \in L$ such that $\|\mathbf{u}\| \leq f\lambda_1(L)$.

Problem 3 (Hermite Shortest Vector Problem (HSVP) [31]). Given a basis of a d-rank integral lattice L and an approximation factor $f > 0$, find a nonzero $\mathbf{u} \in L$ such that $\|\mathbf{u}\| \leq f\mathrm{vol}(L)^{1/d}$.

When $f = 1$, ASVP is exactly SVP. As opposed to SVP and ASVP, it is possible to easily check a solution to HSVP: indeed, given $\mathbf{u}$, L and f, one can check in polynomial time whether or not $\mathbf{u} \in L$ and $\|\mathbf{u}\| \leq f\mathrm{vol}(L)^{1/d}$. By definition of Hermite's constant, if one can solve ASVP with an approximation factor f, then one can solve HSVP with a factor $f\sqrt{\gamma_d}$. Reciprocally, it was shown in [32] that if one has access to an oracle solving HSVP with a factor f, then one can solve ASVP with a factor f^2 in polynomial time using a number of oracle queries linear in the dimension d. Hence, solving ASVP with an approximation factor polynomial in the dimension is equivalent to solving HSP with an approximation factor polynomial in the dimension.

Hardness results for SVP are surveyed in [9, 33], so let us just briefly summarize. SVP was conjectured NP-hard as early as 1981 [34] (see also [32]). Ajtai showed NP-hardness under randomized reductions in 1998 [35], but the historical conjecture with deterministic reductions remains open. The best result so far [12] suggests that it is unlikely that one can efficiently approximate SVP to within quasi-polynomial factors. But NP-hardness results have limits: essentially, approximating SVP within a factor $\sqrt{d/\log d}$ is unlikely to be NP-hard. More precisely, Aharonov and Regev [37] showed that there exists a constant c such that approximating SVP with a factor $c\sqrt{d}$ is in the l′ intersection NP∩coNP, while Goldreich and Goldwasser [38] showed that each constant c approximating SVP with a factor $c\sqrt{d/\log d}$ is in NP∩coAM.

We will present the main algorithms for solving SVP, either exactly or approximately, but we can already summarize the situation. The LLL algorithm [16] (section "The LLL Algorithm") solves ASVP with factor $(4/3+\varepsilon)^{(d-1)/2}$, and HSVP with factor $(4/3+\varepsilon)^{(d-1)/4}$, in time polynomial in $1/\varepsilon$ and the size of the lattice basis. This algorithm is used in the best exact-SVP algorithms:

- Kannan's deterministic algorithm [10] has super-exponential complexity $2^{O(d \log d)}$ polynomial-time operations (see [12] for a tight analysis of the constant).
- The randomized algorithm of Ajtai et al. [14] has exponential complexity $2^{O(d)}$ polynomial-time operations.

The best polynomial-time algorithms known to approximate SVP (better than LLL) are blockwise algorithms that use such exact-SVP algorithms in low dimension: indeed, in dimension d, one can use a subroutine an exact-SVP algorithm in dimension $k = f(d)$, if the function $f(d)$ is sufficiently small that the cost of the subroutine remains polynomial in d. For instance, the super-exponential running-time $2^{O(k \log k)}$ of Kannan's algorithm [10] remains polynomial in d if we select $k = \log d / \log\log d$.

With a number of calls to the SVP-oracle in dimension $\leq k$, Schnorr [17] showed one could approximate SVP with a factor $(2k)^{2d/k}$ and HSVP with a factor $(2k)^{d/k}$. Gama et al. [18] proved that Schnorr's analysis [17] was not optimal: one can raise to the power $\ln 2 \approx 0.69 < 1$ both approximation factors. Gama et al. [18] also presented a slightly better variant: it can approximate SVP with a factor $O(k)^{d/k}$ and HSVP with a factor $O(k)^{d/(2k)}$, still with a polynomial number of calls to the SVP-oracle in dimension $\leq k$. The best blockwise algorithm known is Gama–Nguyen's slide algorithm [19], which approximates SVP with a factor $((1+\varepsilon)\gamma_d)^{(d-k)/(k-1)}$ and HSVP with a factor $\sqrt{(1+\varepsilon)\gamma_d}^{(d-1)/(k-1)}$, with a polynomial (in $1/\varepsilon$ and the size of the lattice basis) number of calls to a SVP-oracle in dimension $\leq k$. When k is fixed, the approximation factors of all these blockwise algorithms remain exponential in d, like for LLL. But if one takes $k = \log d$ and use the AKS algorithm [14] as a SVP-subroutine, one obtains a randomized polynomial-time algorithm approximating SVP and HSP with slightly sub-exponential factors: $2^{O(d \log\log d/\log d)}$.

The Closest Vector Problem

The closest vector problem can be viewed as a homogeneous problem: one is looking for the radius of the smallest hyperball (centered at zero) intersecting the lattice nontrivially. One obtains a nonhomogeneous version by considering hyperballs centered at any point of the space, rather than zero. For any point $\mathbf{x}$ of $\mathbb{R}^n$, and a lattice L of $\mathbb{R}^n$, we will thus denote by $\mathrm{dist}(\mathbf{x}, L)$ the minimal distance between $\mathbf{x}$ and a lattice vector of L. The corresponding computational problem is the following:

Problem 4 (Closest Vector Problem (CVP)). Given a basis of a d-rank integer lattice $L \subseteq \mathbb{Z}^n$, and a point $\mathbf{x} \in \mathbb{Z}^n$, find $\mathbf{y} \in L$ such that $\|\mathbf{x} - \mathbf{y}\| = \mathrm{dist}(\mathbf{x}, L)$.

Similarly to SVP/ASVP, one can define the following approximate version:

Problem 5 (Approximate Closest Vector Problem (ACVP)). Given a basis of a d-rank integer lattice $L \subseteq \mathbb{Z}^n$, a point $\mathbf{x} \in \mathbb{Z}^n$, and an approximation factor $f \geq 1$, find $\mathbf{y} \in L$ such that $\|\mathbf{x} - \mathbf{y}\| \leq f \times \mathrm{dist}(\mathbf{x}, L)$.

In this article, we will not further discuss CVP: we only survey SVP algorithms.

The Two-Dimensional Case

Lagrange's Reduction and Hermite's Constant in Dimension Two

Lagrange [5] formalized for the first time a reduction notion for rank-two lattices, in the language of quadratic forms. This reduction notion is so natural that all other reduction notions usually match in dimension two.

Definition 17. Let L be a two-rank lattice of $\mathbb{R}^n$. A basis $(\mathbf{b}_1, \mathbf{b}_2)$ of L is said to be *Lagrange-reduced* (or simply *L-reduced*) if and only if $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|$ and $|\langle \mathbf{b}_1, \mathbf{b}_2 \rangle| \leq \|\mathbf{b}_1\|^2/2$.

Geometrically, this means that $\mathbf{b}_2$ is inside the disc of radius $\|\mathbf{b}_1\|$ centered at the origin, and that the angle $(\mathbf{b}_1, \mathbf{b}_2)$ modulo π is between $\pi/3$ and $2\pi/3$. Note that the second condition $|\langle \mathbf{b}_1, \mathbf{b}_2 \rangle| \leq \|\mathbf{b}_1\|^2/2$ is equivalent to size-reduction.

The definition implies that it is trivial to check whether a given basis is L-reduced or not. The following result shows that this reduction notion is optimal in a natural sense:

Theorem 7. *Let* $(\mathbf{b}_1, \mathbf{b}_2)$ *be a basis of a two-rank lattice* L *of* $\mathbb{R}^n$. *The basis* $(\mathbf{b}_1, \mathbf{b}_2)$ *is Lagrange-reduced if and only if* $\|\mathbf{b}_1\| = \lambda_1(L)$ *and* $\|\mathbf{b}_2\| = \lambda_2(L)$.

Assuming this result, it is clear that there always exist L-reduced bases. And by definition, the first vector of any such basis satisfies

$$\|\mathbf{b}_1\| \leq (4/3)^{1/4} \mathrm{vol}(L)^{1/2}.$$

In particular, one can deduce the inequality $\gamma_2 \leq \sqrt{4/3}$. But one also knows that $\gamma_2 \geq \sqrt{4/3}$, by considering the hexagonal lattice spanned by $(\mathbf{b}_1, \mathbf{b}_2)$ such that $\|\mathbf{b}_1\| = \|\mathbf{b}_2\|$ and $\langle \mathbf{b}_1, \mathbf{b}_2 \rangle = \|\mathbf{b}_1\|^2/2$, which is the equality case of Lagrange's reduction.

In other words, one can arguably summarize Lagrange's reduction by a single equality

$$\gamma_2 = \sqrt{4/3}.$$

Lagrange's Algorithm

Lagrange's algorithm [5] solves the two-rank lattice reduction problem: it finds a basis achieving the first two minima, in a running time similar to Euclid's algorithm. It is often incorrectly attributed to Gauss [6]. Lagrange's algorithm can be viewed as a two-dimensional generalization of the centered variant of Euclid's algorithm (Algorithm 1).

```
Input: $(n, m) \in \mathbb{Z}^2$.
Output: $\gcd(n, m)$.
 1:
 2: if $|n| \leq |m|$ then
 3:     swap $n$ and $m$.
 4: end if
 5:
 6: while $m \neq 0$ do
 7:     $r \longleftarrow n - qm$ where $q = \left\lfloor \frac{n}{m} \right\rceil$.
 8:     $n \longleftarrow m$
 9:     $m \longleftarrow r$
10: end while
11: Output $|n|$.
```

Algorithm 1: Euclid's centered algorithm

This algorithm corresponds to a reduction in dimension one. Indeed, the gcd is simply the first minimum of the lattice $n\mathbb{Z} + m\mathbb{Z}$ spanned by n and m. The only difference with the classical Euclidean algorithm is in Step 7, where one takes for q the closest integer to $\frac{n}{m}$, rather than its integral part. This amounts to selecting the integer q to minimize $|n - qm|$, which guarantees $|n - qm| \leq \frac{|m|}{2}$. It is easy to show that Euclid's centered algorithm has quadratic complexity without fast integer arithmetic.

Lagrange's algorithm (Algorithm 2) is a natural generalization in dimension two.

```
Input: a basis $(\mathbf{u}, \mathbf{v})$ of a two-rank lattice $L$.
Output: an L-reduced basis of $L$, reaching $\lambda_1(L)$ and $\lambda_2(L)$.
1: if $\|\mathbf{u}\| < \|\mathbf{v}\|$ then
2:     swap $\mathbf{u}$ and $\mathbf{v}$
3: end if
4: repeat
5:     $\mathbf{r} \longleftarrow \mathbf{u} - q\mathbf{v}$ where $q = \left\lfloor \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{v}\|^2} \right\rceil$.
6:     $\mathbf{u} \longleftarrow \mathbf{v}$
7:     $\mathbf{v} \longleftarrow \mathbf{r}$
8: until $\|\mathbf{u}\| \leq \|\mathbf{v}\|$
9: Output $(\mathbf{u}, \mathbf{v})$.
```

Algorithm 2: Lagrange's reduction algorithm

The analogy is clear: Step 5 selects the integer q such that $\mathbf{r} = \mathbf{u} - q\mathbf{v}$ is as short as possible. This is precisely the case when the orthogonal projection of $\mathbf{r}$ over $\mathbf{v}$ is as short as possible, and this projection can have length less than $\leq \|\mathbf{v}\|/2$. This can be viewed geometrically, and an elementary computation shows that $q = \left\lfloor \frac{\langle \mathbf{u},\mathbf{v}\rangle}{\|\mathbf{v}\|^2} \right\rceil$ works.

One can show that Lagrange's algorithm has quadratic complexity (in the maximal bit-length of the coefficients of the input basis) without fast integer arithmetic, see [39]. For further generalizations of Lagrange's algorithm, see [39,40].

Gram–Schmidt Orthogonalization and Size-Reduction

If $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{Z}^n$ have norms bounded by B, the computation of all Gram–Schmidt coefficients (i.e., of the rational numbers $\mu_{i,j}$ and $\|\mathbf{b}_i^\star\|^2$) can be done in time $O(d^5 \log^2 B)$ without fast arithmetic.

From the triangular representation of the basis, it is very easy to see how to size-reduce a basis (See Algorithm 3): the vectors $\mathbf{b}_i$'s are modified, but not their projections $\mathbf{b}_i^\star$.

```
Input: A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice $L$.
Output: A size-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$.
Compute all the Gram–Schmidt coefficients $\mu_{i,j}$.
for $i = 2$ to $d$ do
    for $j = i - 1$ downto 1 do
        $\mathbf{b}_i \longleftarrow \mathbf{b}_i - \lceil \mu_{i,j} \rfloor \mathbf{b}_j$
        for $k = 1$ to $j$ do
            $\mu_{i,k} \longleftarrow \mu_{i,k} - \lceil \mu_{i,j} \rfloor \mu_{j,k}$
        end for
    end for
end for
```

Algorithm 3: A size-reduction algorithm

Hermite's Inequality and the Lenstra–Lenstra–Lovász Algorithm

All the algorithms of this section can be viewed as algorithmic versions of the following elementary result:

Theorem 8 (Hermite's inequality [7]). *For all integer $d \geq 2$,*

$$\gamma_d \leq \gamma_2^{d-1}. \tag{2.21}$$

Proof. We give a proof by induction, slightly different from the historical proof of Hermite. As the inequality is trivial for $d = 2$, assume that it holds for $d - 1$. Consider a shortest nonzero vector $\mathbf{b}_1$ of a d-rank lattice L. Denote by $L' = \pi_2(L)$ the $(d-1)$-rank lattice obtained by projecting L over $\mathbf{b}_1^{\perp}$. Its volume is $\mathrm{vol}(L') = \mathrm{vol}(L)/\|\mathbf{b}_1\|$. Let $\mathbf{b}'_2$ be a shortest nonzero vector of L'. The induction assumption ensures that

$$\|\mathbf{b}'_2\| \leq (4/3)^{(d-2)/4}\mathrm{vol}(L')^{1/(d-1)}.$$

We can lift $\mathbf{b}'_2$ (by size-reduction) into a nonzero vector $\mathbf{b}_2 \in L$ such that $\|\mathbf{b}_2\|^2 \leq \|\mathbf{b}'_2\|^2 + \|\mathbf{b}_1\|^2/4$. As $\mathbf{b}_1$ cannot be longer than $\mathbf{b}_2$, we deduce

$$\|\mathbf{b}_1\| \leq \sqrt{4/3}\|\mathbf{b}'_2\| \leq (4/3)^{d/4}\mathrm{vol}(L')^{1/(d-1)},$$

which can be rewritten as

$$\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d},$$

which completes the proof. In retrospect, one notices that with the inequality $\|\mathbf{b}_1\| \leq \sqrt{4/3}\|\mathbf{b}^{\star}_2\|$, one has in fact proved the inequality

$$\gamma_d \leq (4\gamma_{d-1}/3)^{(d-1)/d}.$$

By composing all these inequalities, one indeed obtains Hermite's inequality

$$\gamma_d \leq (4/3)^{(d-1)/d+(d-2)/d+\cdots+1/d} = (4/3)^{(d-1)/2}.$$

The historical proof given by Hermite in his first letter [7] to Jacobi also proceeded by induction, but in a slightly different way. Hermite considered an arbitrary primitive vector $\mathbf{b}_1$ of the lattice L. If $\mathbf{b}_1$ satisfies Hermite's inequality, that is, if $\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d}$, there is nothing to prove. Otherwise, one applies the induction assumption to the projected lattice $L' = \pi_2(L)$: one knows that there exists a primitive vector $\mathbf{b}^{\star}_2 \in L'$ satisfying Hermite's inequality: $\|\mathbf{b}^{\star}_2\| \leq (4/3)^{(d-2)/4}\mathrm{vol}(L')^{1/(d-1)}$. One can lift this vector $\mathbf{b}^{\star}_2 \in L'$ into a primitive vector $\mathbf{b}_2 \in L$ such that $\|\mathbf{b}_2\|^2 \leq \|\mathbf{b}^{\star}_2\|^2 + \|\mathbf{b}_1\|^2/4$. As $\mathbf{b}_1$ does not satisfy Hermite's inequality, one notices that $\|\mathbf{b}_2\| < \|\mathbf{b}_1\|$: one can therefore replace $\mathbf{b}_1$ by $\mathbf{b}_2$ and start again. But this process cannot go on indefinitely: indeed, there are only finitely many vectors of L that have norm $\leq \|\mathbf{b}_1\|$. Hence, there must exist a nonzero vector $\mathbf{b}_1 \in L$ satisfying Hermite's inequality. □

The inequality (2.21) suggests to use two-dimensional reduction to find in any d-rank lattice a nonzero vector of norm less than

$$\sqrt{\gamma_2^{d-1}}\mathrm{vol}(L)^{1/d} = (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d}.$$

This is somewhat the underlying idea behind all the algorithms of this section: Hermite's algorithms and the LLL algorithm. In fact, the proof of (2.21) that we gave provides such an algorithm, implicitly. This algorithm makes sure that the basis is size-reduced and that all the local bases $(\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1})) = (\mathbf{b}_i^\star, \mathbf{b}_{i+1}^\star + \mu_{i+1,i}\mathbf{b}_i^\star)$ are L-reduced: these local bases correspond to the 2×2 matrices on the diagonal, when we represent the basis in triangular form. In other words, the reduced bases obtained are size-reduced and such that for all $1 \leq i \leq d$:

$$\|\mathbf{b}_{i+1}^\star\|^2 \geq \frac{3}{4}\|\mathbf{b}_i^\star\|^2, \tag{2.22}$$

that is, the decrease of the norms of the Gram–Schmidt vectors (which are the diagonal coefficients in the triangular representation) is at most geometric, which is sometimes called Siegel's condition [2]. It is then easy to see that the first vector of such a basis satisfies

$$\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d},$$

as announced. But it is unknown if this algorithm and those of Hermite are polynomial time: the LLL algorithm guarantees a polynomial running-time by relaxing inequalities (2.22).

Hermite's Algorithms

We now describe the first reduction algorithms in arbitrary dimension, described by Hermite in his famous letters [7] to Jacobi, in the language of quadratic forms. They are very close to the algorithm underlying the proof of (2.21), but they do not explicitly rely on Lagrange's algorithm, although they try to generalize it. They were historically presented in a recursive way, but they can easily be made iterative, just like LLL.

```
Input: A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a $d$-rank lattice $L$.
Output:
if $d = 1$ then
    output $\mathbf{b}_1$
end if
Apply recursively the algorithm to the basis $(\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_d))$ of the projected lattice $\pi_2(L)$.
Lift the vectors $(\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_d))$ into $\mathbf{b}_2, \ldots, \mathbf{b}_d \in L$ in such a way that they are size-reduced with respect to $\mathbf{b}_1$.
if $\mathbf{b}_1$ satisfies Hermite's inequality, that is $\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d}$ then
    Output $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$
end if
Swap $\mathbf{b}_1$ and $\mathbf{b}_2$ since $\|\mathbf{b}_2\| < \|\mathbf{b}_1\|$, and restart from the beginning.
```

Algorithm 4: A simplified version of Hermite's first reduction algorithm, described in the first letter to Jacobi [7]

Hermite's first algorithm was described in the first letter [7] to Jacobi: Algorithm 4 is a simplified version of this algorithm; Hermite's historical algorithm actually uses duality, which we ignore for simplicity. It is easy to see that Algorithm 4 terminates, and that the output basis $(\mathbf{b}_1, \dots, \mathbf{b}_d)$ satisfies the following reduction notion (which we call H1):

- The basis is size-reduced.
- For all i, $\mathbf{b}_i^\star$ verifies Hermite's inequality in the projected lattice $\pi_i(L)$:

$$\|\mathbf{b}_i^\star\| \leq (4/3)^{(d-i)/4} \mathrm{vol}(\pi_i(L))^{1/(d-i+1)}$$

Notice that this reduction notion is rather weak: for instance, the orthogonality defect of a H1-reduced basis can be arbitrarily large, as soon as the dimension is greater than 3, as shown by the following triangular basis (where $\varepsilon > 0$ tends to 0):

$$\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & \varepsilon & 0 \\ 1/2 & \varepsilon/2 & 1/\varepsilon \end{pmatrix}.$$

By the way, Hermite notices himself that his first algorithm does not match with Lagrange's algorithm in dimension two. It seems to be one of the reasons why he presents a second algorithm (Algorithm 5) in his second letter [7] to Jacobi.

```
Input: a basis $(\mathbf{b}_1, \dots, \mathbf{b}_d)$ of a lattice $L$.
Output: a size-reduced basis $(\mathbf{b}_1, \dots, \mathbf{b}_d)$ such that for all $i$, $\|\mathbf{b}_i^\star\|/\|\mathbf{b}_{i+1}^\star\| \leq \gamma_2 = \sqrt{4/3}$. In
    particular, each $\mathbf{b}_i^\star$ satisfies Hermite's inequality in the projected lattice $\pi_i(L)$.
if $d = 1$ then
    output $\mathbf{b}_1$
end if
By making swaps if necessary, ensure that $\|\mathbf{b}_1\| \leq \|\mathbf{b}_i\|$ for all $i \geq 2$.
Apply recursively the algorithm to the basis $(\pi_2(\mathbf{b}_2), \dots, \pi_2(\mathbf{b}_d))$ of the projected lattice
    $\pi_2(L)$.
Lift the vectors $(\pi_2(\mathbf{b}_2), \dots, \pi_2(\mathbf{b}_d))$ to $\mathbf{b}_2, \dots, \mathbf{b}_d \in L$ in such a way that they are size-
    reduced with respect to $\mathbf{b}_1$.
if $\|\mathbf{b}_1\| \leq \|\mathbf{b}_i\|$ for all $i \geq 2$ then
    output $(\mathbf{b}_1, \dots, \mathbf{b}_d)$
else
    restart from the beginning.
end if
```

Algorithm 5: Hermite's second reduction algorithm, described in his second letter to Jacobi [7]

It is easy to see that this algorithm terminates and that the output basis $(\mathbf{b}_1, \dots, \mathbf{b}_d)$ satisfies the following reduction notion (which we call H2):

- The basis is size-reduced.
- For all i, $\mathbf{b}_i^\star$ has minimal norm among all the vectors of the basis $(\pi_i(\mathbf{b}_i), \pi_i (\mathbf{b}_{i+1}) \dots, \pi_i(\mathbf{b}_d))$ of the projected lattice $\pi_i(L)$, that is $\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$ for all $1 \leq i \leq j \leq d$.

Notice that an H2-reduced basis necessarily satisfies (2.22), that is, for all i

$$\|\mathbf{b}_i^\star\|/\|\mathbf{b}_{i+1}^\star\| \leq \gamma_2 = \sqrt{4/3}.$$

This implies that its orthogonality defect is bounded:

$$\prod_{i=1}^{d} \|\mathbf{b}_i^\star\| \leq (4/3)^{d(d-1)/4} \mathrm{vol}(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_d)).$$

And this also shows that an H2-reduced basis is necessarily H1-reduced.

Hermite's second algorithm is very close to the so-called deep insertion variant of LLL by Schnorr and Euchner [41]: both algorithms want to achieve the same reduction notion.

The LLL Algorithm

Surprisingly, it is unknown if Hermite's algorithms are polynomial time for varying dimension. It is also the case for Lenstra's algorithm [42], which is a relaxed variant of Hermite's second algorithm, where the inequalities $\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$ are replaced by $c\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$, where c is a constant such that $1/4 < c < 1$. However, Lenstra proved that his algorithm was polynomial time for any fixed dimension, which was sufficient for his celebrated result on integer programming [42].

It is Lenstra et al. [16] who invented in 1982 the first polynomial-time reduction algorithm outputting basis nearly as reduced as Hermite's. This algorithm, known as LLL or L^3, is essentially a relaxed variant of Hermite's second algorithm: László Lovász discovered that a crucial modification guaranteed a polynomial running-time; more precisely, compared to the H2 reduction notion, one replaces for each i all the inequalities $\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$ by a single inequality $c\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_{i+1})\|$, where c is a constant such that $1/4 < c < 1$. The final algorithm was published in [16].

Let δ be a real in $[\frac{1}{4}, 1]$. A numbered basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L is said to be *LLL-reduced* with factor δ if it is size-reduced, and if it satisfies *Lovász' condition*: for all $1 < i \leq d$,

$$\left\|\mathbf{b}_{i+1}^\star + \mu_{i+1,i}\mathbf{b}_i^\star\right\|^2 \geq \delta\|\mathbf{b}_i^\star\|^2.$$

Let us explain this mysterious condition. As Gram–Schmidt orthogonalization depends on the order of the vectors, its vectors change if $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$ are swapped; in fact, only $\mathbf{b}_i^\star$ and $\mathbf{b}_{i+1}^\star$ can possibly change. And the new $\mathbf{b}_i^\star$ is simply $\mathbf{b}_{i+1}^\star + \mu_{i+1,i}\mathbf{b}_i^\star$; therefore, Lovász' condition means that by swapping $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$, the norm of $\mathbf{b}_i^\star$ does not decrease too much, where the loss is quantified by δ: one cannot gain much on $\|\mathbf{b}_i^\star\|$ by swap. In other words,

$$\delta \|\mathbf{b}_i^\star\|^2 \leq \|\pi_i(\mathbf{b}_{i+1})\|^2,$$

which illustrates the link with the H2 reduction notion. The most natural value for the constant δ is therefore $\delta = 1$ (in dimension 2, this matches with Lagrange's reduction), but then, it is unknown if such a reduced basis can be computed in polynomial time. The LLL-reduction was initially[1] presented in [16] with the factor $\delta = \frac{3}{4}$, so that in the literature, LLL-reduction usually means LLL-reduction with the factor $\delta = \frac{3}{4}$.

Lovász' condition can also be rewritten equivalently: for all i,

$$\|\mathbf{b}_{i+1}^\star\|^2 \geq \left(\delta - \mu_{i+1,i}^2\right) \|\mathbf{b}_i^\star\|^2,$$

which is a relaxation of (2.22). Thus, LLL reduction guarantees that each $\mathbf{b}_{i+1}^\star$ cannot be much shorter than $\mathbf{b}_i^\star$: the decrease is at most geometric. This proves the following result:

Theorem 9. *Assume that $\frac{1}{4} < \delta \leq 1$, and let $\alpha = 1/(\delta - \frac{1}{4})$. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be an LLL-reduced basis with factor δ of a lattice L in $\mathbb{R}^n$. Then*

1. $\|\mathbf{b}_1\| \leq \alpha^{(d-1)/4}(\mathrm{vol} L)^{1/d}$.
2. *For all $i \in \{1, \ldots, d\}$, $\|\mathbf{b}_i\| \leq \alpha^{(d-1)/2}\lambda_i(L)$.*
3. $\|\mathbf{b}_1\| \times \cdots \times \|\mathbf{b}_d\| \leq \alpha^{d(d-1)/4} \det L$.

Thus, an LLL-reduced basis provides an approximation of the lattice reduction problem. By taking δ very close to 1, one falls back on Hermite's inequality in an approximate way, where the constant $4/3$ is replaced by $4/3 + \varepsilon$.

The other interest of this reduction notion is that there exists a simple algorithm to compute such reduced bases, and which is rather close to Hermite's second algorithm (Algorithm 5). In its simplest form, the LLL algorithm corresponds to Algorithm 6.

```
Input: a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice $L$.
Output: the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is LLL-reduced with factor $\delta$.
1: Size-reduce $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ (using Algorithm 3).
2: if there exists an index $j$ which does not satisfy Lovász' condition then
3:     swap $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$, then return to Step 1.
4: end if
```

Algorithm 6: The basic LLL algorithm

Compared to this simple version, the so-called iterative versions of the LLL algorithm consider instead the smallest index j not satisfying Lovász' condition: in contrast, Hermite's second algorithm considered the greatest index j refuting H2.

[1] This simplifies the exposition.

Theorem 10. *Assume that $\frac{1}{4} < \delta < 1$. If each $\mathbf{b}_i \in \mathbb{Q}^n$, Algorithm 6 computes an LLL-reduced basis in time polynomial in the maximal bit-length of the coefficients of the $\mathbf{b}_i$'s, the lattice rank d, and the space dimension n.*

Let us sketch a proof of this fundamental result, assuming to simplify that $\mathbf{b}_i \in \mathbb{Z}^n$. First of all, it is clear that if the algorithm terminates, then the output basis is LLL-reduced with factor δ. To see why the algorithm terminates, let us analyze each swap (Step 3). When $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$ are swapped, only $\mathbf{b}_j^\star$ and $\mathbf{b}_{j+1}^\star$ can be modified among all the Gram–Schmidt vectors. Let us therefore denote by $\mathbf{c}_j^\star$ and $\mathbf{c}_{j+1}^\star$ the new Gram–Schmidt vectors after swapping. As the product of all the Gram–Schmidt vector norms is equal to $\mathrm{vol}(L)$, we have

$$\|\mathbf{c}_j^\star\| \times \|\mathbf{c}_{j+1}^\star\| = \|\mathbf{b}_j^\star\| \times \|\mathbf{b}_{j+1}^\star\|.$$

As Lovász' condition is not satisfied, $\|\mathbf{c}_j^\star\|^2 < \delta\|\mathbf{b}_j^\star\|^2$. Hence,

$$\|\mathbf{c}_j^\star\|^{2(d-j+1)}\|\mathbf{c}_{j+1}^\star\|^{2(d-j)} < \delta\|\mathbf{b}_j^\star\|^{2(d-j+1)}\|\mathbf{b}_{j+1}^\star\|^{2(d-j)}.$$

This suggests to consider the following quantity:

$$D = \|\mathbf{b}_1^\star\|^{2d}\|\mathbf{b}_2^\star\|^{2(d-1)} \times \cdots \times \|\mathbf{b}_d^\star\|^2.$$

At each swap, D decreases by a factor $\delta < 1$. Notice that D can be decomposed as a product of d Gram determinants $D_i = \Delta(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ for i going through 1 to d. Therefore, D is in fact an integer, as $\mathbf{b}_i \in \mathbb{Z}^n$. It follows that the number of swaps is at most logarithmic in the initial value of D, which can be upper bounded by B^{2d}, where B is the maximum of the initial norms $\|\mathbf{b}_i\|$. To bound the complexity of the algorithm, one also needs to upper bound the size of the rational coefficients $\mu_{i,j}$ and $\|\mathbf{b}_i^\star\|^2$ during the reduction. A careful analysis based on the D_i's shows that all the $\mu_{i,j}$'s always have polynomial size (see [16,32,43,44]).

By coupling Theorem 9 with Theorem 10, we can summarize the LLL result as follows:

Corollary 4. *There exists an algorithm which, given as input a basis of a d-dimensional integer lattice $L \subseteq \mathbb{Z}^n$ and a reduction factor $\varepsilon > 0$, outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L, in time polynomial in $1/\varepsilon$ and the size of the basis, such that*

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \le \left((1+\varepsilon)\sqrt{4/3}\right)^{(d-1)/2},$$

$$\|\mathbf{b}_i\|/\lambda_i(L) \le \left((1+\varepsilon)\sqrt{4/3}\right)^{d-1}, \quad 1 \le i \le d,$$

$$\left(\prod_{i=1}^{d}\|\mathbf{b}_i\|\right)/\mathrm{vol}(L) \le \left((1+\varepsilon)\sqrt{4/3}\right)^{d(d-1)/2}.$$

Solving Exact SVP

In this section, we survey the two main algorithms for finding the shortest vector in a lattice: enumeration [10,45,46] and sieving [14], which both use the LLL algorithm in their first stage. In section "Mordell's Inequality and Blockwise Algorithms", we use such algorithms in low dimension as subroutines to obtain polynomial-time algorithms with better approximation factors than LLL.

Enumeration Algorithms

The simplest method consists in enumerating the coordinates of a shortest lattice vector, and this idea goes back to the early 1980s with Pohst [45], Kannan [10], and Fincke-Pohst [46]. More precisely, by using LLL-reduced bases or other reduced bases not far from being orthogonal, it is possible to exhaustive search the projections of any shortest vector in the projected lattices $\pi_i(L)$.

Consider a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L. Let $\mathbf{x} \in L$ be a (nonzero) shortest vector of L: $\mathbf{x} = x_1\mathbf{b}_1 + \cdots + x_d\mathbf{b}_d$, where the x_i's are integers. We have

$$\mathbf{x} = \sum_{i=1}^{d} x_i \mathbf{b}_i = \sum_{i=1}^{d} x_i \left(\mathbf{b}_i^\star + \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^\star \right) = \sum_{j=1}^{d} \left(x_j + \sum_{i=j+1}^{d} \mu_{i,j} x_i \right) \mathbf{b}_j^\star.$$

It follows that the projections of $\mathbf{x}$, together with their norms, are given by

$$\pi_k(\mathbf{x}) = \sum_{j=k}^{d} \left(x_j + \sum_{i=j+1}^{d} \mu_{i,j} x_i \right) \mathbf{b}_j^\star, \; 1 \leq k \leq d, \tag{2.23}$$

$$\|\pi_k(\mathbf{x})\|^2 = \sum_{j=k}^{d} \left(x_j + \sum_{i=j+1}^{d} \mu_{i,j} x_i \right)^2 \|\mathbf{b}_j^\star\|^2, \; 1 \leq k \leq d. \tag{2.24}$$

Now, let B be an upper bound on $\lambda_1(L) = \|\mathbf{x}\|$, we take $B = \sqrt{\gamma_d}\mathrm{vol}(L)^{1/d}$, but we could also have taken $B = \|\mathbf{b}_1\|$; if ever one knows a better upper bound B, which might be the case for special lattices, then this will decrease the running time of enumeration. Using (2.24), the d inequalities $\|\pi_k(\mathbf{x})\| \leq B$ enable us to exhaustive search of the coordinates $x_d, x_{d-1}, \ldots, x_1$ of $\mathbf{x}$:

$$\sum_{j=k}^{d} \left(x_j + \sum_{i=j+1}^{d} \mu_{i,j} x_i \right)^2 \|\mathbf{b}_j^\star\|^2 \leq B^2, \; 1 \leq k \leq d,$$

which can be rewritten as

$$\left| x_k + \sum_{i=k+1}^{d} \mu_{i,j} x_i \right| \leq \frac{\sqrt{B^2 - \sum_{j=k+1}^{d} \left(x_j + \sum_{i=j+1}^{d} \mu_{i,j} x_i \right)^2 \|\mathbf{b}_j^\star\|^2}}{\|\mathbf{b}_k^\star\|}, \quad 1 \leq k \leq d. \tag{2.25}$$

We start with (2.25), with $k = d$, that is, $|x_d| \leq B/\|\mathbf{b}_d^\star\|$. This allows to exhaustive search of the integer x_d. Now assume that the projection $\pi_{k+1}(\mathbf{x})$ has been guessed for some k: the integers $x_{k+1}, \ldots, x_d$ are known. Then (2.25) enables to compute an interval I_k such that $x_k \in I_k$, and therefore to exhaustive search x_k. For a full description of an exact algorithm implementing this exhaustive search, we refer to [41].

Rigorous Upper Bounds

We start with an elementary result:

Lemma 9. *Let* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ *be an LLL-reduced basis and* $B = \|\mathbf{b}_1\|$. *Then for each* $(x_{k+1}, \ldots, x_d) \in \mathbb{Z}^{d-k}$, *the number of* $x_k \in \mathbb{Z}$ *satisfying* (2.25) *is at most*

$$\lfloor 2\|\mathbf{b}_1\|/\|\mathbf{b}_k^\star\| \rfloor + 1 = 2^{O(k)}.$$

This implies that if $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is an LLL-reduced basis and $B = \|\mathbf{b}_1\|$, then the cost of enumeration is, up to a polynomial-time multiplicative factor,

$$\prod_{k=1}^{d} 2^{O(k)} = 2^{O(d^2)}.$$

Kannan [10–12] showed how to decrease $2^{O(d^2)}$ to $2^{O(d \log d)}$ using a stronger reduction notion than LLL, close to HKZ-reduction. More precisely, Kannan used quasi-HKZ-reduction, which means that $(\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_d))$ is HKZ-reduced, and that $\|\mathbf{b}_1\|$ is not much longer than $\|\mathbf{b}_2^\star\|$. And Kannan [10] noticed that by applying recursively the enumeration algorithm, one could transform an LLL-reduced basis into a quasi-HKZ-reduced basis in $2^{O(d \log d)}$ polynomial-time operations. Kannan [10]'s recursive enumeration algorithm has therefore a total complexity of $2^{O(d \log d)}$ polynomial-time operations. Recently, Hanrot and Stehlé [12, 13] showed that the worst-case complexity of Kannan's algorithm is $d^{d/(2e)+o(d)}$ polynomial-time operations.

Unfortunately, the practical interest of Kannan's algorithm is unclear. More precisely, Nguyen and Vidick [15] provides experimental evidence that for dimensions of practical interest, the $2^{O(d \log d)}$ polynomial-time operations of Kannan [10] are much slower than the $2^{O(d^2)}$ polynomial-time operations of basic enumeration from an LLL-reduced basis. This can be explained as follows: in both cases, the polynomial-time operations and the $O()$ constants are not the same.

Heuristic Estimates

The previous analysis gave only upper bounds. To provide an intuition on the exact cost of enumeration, we now give a heuristic analysis. The cost of enumeration is $\sum_{k=1}^{d} N_k$ up to a multiplicative polynomial-time factor, where N_k is the number of $(x_k, \ldots, x_d) \in \mathbb{Z}^{d-k+1}$ satisfying (2.25). Thus, N_k is exactly the number of vectors in $\pi_k(L)$ of norm $\leq B$. By the Gaussian heuristic (see Definition 8 of section "Volume and the Gaussian Heuristic"), we hope that $N_k \approx H_k$ defined by

$$H_k = \frac{B^{d-k+1} v_{d-k+1}}{\mathrm{vol}(\pi_k(L))} = \frac{B^{d-k+1} v_{d-k+1} \mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_{k-1})}{\mathrm{vol}(L)} \tag{2.26}$$

Let us try to estimate (2.26) for typical reduced bases. It has been reported (see [31,47]) that for most practical reduction algorithms in high dimension, except when the lattice has a very special structure, applying the reduction algorithm to a sufficiently randomized input basis gives rise to a reduced basis such that $\|\mathbf{b}_i^\star\| / \|\mathbf{b}_{i+1}^\star\| \approx q$, where q depends on the algorithm:

- for LLL, $q \approx 1.02^2 \approx 1.04$ in high dimension.
- for BKZ-20 [41], $q \approx 1.025$.

It follows that $\|\mathbf{b}_1\| \approx q^{(d-1)/2} \mathrm{vol}(L)^{1/d}$ and

$$\frac{\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_{k-1})}{\mathrm{vol}(L)} \approx \frac{\|\mathbf{b}_1\|^{k-1}}{q^{1+2+\cdots+k-2} \mathrm{vol}(L)} = \frac{\|\mathbf{b}_1\|^{k-1}}{q^{(k-2)(k-1)/2} \mathrm{vol}(L)}.$$

Then (2.26) becomes

$$H_k \approx \frac{B^{d-k+1} v_{d-k+1} \|\mathbf{b}_1\|^{k-1}}{q^{(k-2)(k-1)/2} \mathrm{vol}(L)}. \tag{2.27}$$

The complexity will depend on the choice of the upper bound B:

- If one takes $B = \|\mathbf{b}_1\|$, then (2.27) becomes

$$H_k \approx \frac{\|\mathbf{b}_1\|^d v_{d-k+1}}{q^{(k-2)(k-1)/2} \mathrm{vol}(L)} = \frac{q^{d(d-1)/2} v_{d-k+1}}{q^{(k-2)(k-1)/2}}$$

$$= q^{[d(d-1)-(k-2)(k-1)]/2} v_{d-k+1}$$

Thus,

$$H_k \lessapprox q^{d^2/2+o(d^2)}.$$

- If one takes $B = \sqrt{\gamma_d}\mathrm{vol}(L)^{1/d}$, then $\sqrt{\gamma_d} = \Theta(\sqrt{d})$ implies that (2.27) becomes

$$H_k \approx \frac{\|\mathbf{b}_1\|^{k-1}2^{O(d)}}{q^{(k-2)(k-1)/2}\mathrm{vol}(L)^{(k-1)/d}} = \frac{q^{(k-1)(d-1)/2}2^{O(d)}}{q^{(k-2)(k-1)/2}}$$

$$= q^{(k-1)(d-k+1)/2}2^{O(d)},$$

where the right-hand term is always less than $q^{d^2/8-1/2}2^{O(d)}$, because $(k-1)(d-k+1)$ is maximized for $k = d/2$. Hence,

$$H_k \lessapprox q^{d^2/8}2^{O(d)}.$$

In both cases, $\max_k H_k$ is super-exponential in d, but the exponentiation base ($q^{1/2}$ or $q^{1/8}$) is very close to 1.

A Heuristic Lower Bound

One might wonder if Kannan's worst-case complexity of $d^{d/(2e)+o(d)}$ polynomial-time operations can be improved using a different reduction notion. By definition of Rankin's constant, we have:

$$H_k \geq \frac{B^{d-k+1}v_{d-k+1}\sqrt{\gamma_{d,k-1}(L)}\mathrm{vol}(L)^{(k-1)/d}}{\mathrm{vol}(L)} = \frac{B^{d-k+1}v_{d-k+1}\sqrt{\gamma_{d,k-1}(L)}}{\mathrm{vol}(L)^{d-k+1}}.$$

If we take $B = \sqrt{\gamma_d}\mathrm{vol}(L)^{1/d}$, we obtain

$$H_k \geq \sqrt{\gamma_d}^{d-k+1}v_{d-k+1}\sqrt{\gamma_{d,k-1}(L)}.$$

Now recall that $\sqrt{\gamma_d} = \Theta(\sqrt{d})$, which implies that

$$H_k \geq v_{d-k+1}\Theta(\sqrt{d})^{d-k+1}\sqrt{\gamma_{d,k-1}(L)}.$$

An elementary (but tedious) computation shows that as d grows to infinity, for all $1 \leq k \leq d$,

$$v_{d-k+1}\Theta(\sqrt{d})^{d-k+1} = 2^{\Theta(d)}.$$

Hence:

$$H_k \geq 2^{\Theta(d)}\sqrt{\gamma_{d,k-1}(L)}.$$

But using (2.18) with $m = \lfloor n/2 \rceil$, we know that

$$\max_{k=2}^{d} \gamma_{d,k-1} \geq \Omega(d)^{d/4+o(d)}.$$

Therefore,

$$H_{\lfloor d/2 \rceil} \geq 2^{\Theta(d)} d^{d/8+o(d)}.$$

This suggests that, independently of the quality of the reduced basis, the complexity of enumeration will be at least $d^{d/8}$ polynomial-time operations for many lattices.

Sieve Algorithms

In 2001, Ajtai et al. [14] discovered a randomized algorithm, which is asymptotically much better than Kannan's deterministic super-exponential algorithm [10]. Indeed, the AKS algorithm outputs with overwhelming probability a shortest vector of a lattice L in $2^{O(d)}$ polynomial-time operations. Running time apart, the algorithm is interesting because it is based on totally different principle: sieving.

We just give the main idea, making significant simplifications: for more details, see [14] or [15], which presents the most practical variant known of AKS. This heuristic variant [15] has complexity $(4/3)^d$ polynomial-time operations, but the output is not guaranteed to be a shortest vector.

Consider a ball S centered at the origin and of radius r such that $\lambda_1(L) \leq r \leq O$ $(\lambda_1(L))$. Then $|L \cap S| = 2^{O(d)}$. If we could exhaustive search $L \cap S$, we could output the shortest vector within $2^{O(d)}$ polynomial-time operations. Enumeration algorithms do perform an exhaustive search of $L \cap S$, but to do so, they also require to go through all the points of $\cup_{1 \leq k \leq d} \pi_k(L) \cap S$. Because $\sum_{k=1}^{d} |\pi_k(L) \cap S| = 2^{O(d \log d)}$ in the worst case for HKZ-reduced bases, and the worst-case complexity of Kannan's algorithm is $2^{O(d \log d)}$, rather than $2^{O(d)}$, up to some polynomial-time factor .

The main idea of sieve algorithms is to do a randomized sampling of $L \cap S$, without going through the much larger set $\cup_{1 \leq k \leq d} \pi_k(L) \cap S$. If sampling was such that each point of $L \cap S$ was output with probability roughly $|L \cap S|^{-1}$, and if $N \gg |L \cap S|$, then one of N samples would be a shortest vector with probability close to 1. Unfortunately, it is unclear if this property is satisfied by the AKS sampling. However, it can be shown that there exists $\mathbf{w} \in L \cap S$ such that both $\mathbf{w}$ and $\mathbf{w} + \mathbf{s}$, where $\mathbf{s}$ is a shortest vector, can be output with nonzero probability. Thus, by computing the shortest difference between the N sampled vectors in $L \cap S$, where $N \gg |L \cap S|$, one obtains a shortest vector of L with probability close to 1.

However, sampling directly in a ball centered at 0 and of radius r such that $\lambda_1(L) \leq r \leq O(\lambda_1(L))$ is difficult. But, starting with an LLL-reduced basis, it is easy to sample with a radius $2^{O(d)}\lambda_1(L)$. To decrease the factor $2^{O(d)}$ to $O(1)$, one uses a sieve, which is the most expensive stage of the algorithm.

Sieving iteratively shortens the vectors of S by a geometric factor of at least γ (such that $0 < \gamma < 1$) at each iteration; thus, a linear number of sieve iterations suffices to decrease the multiplicative factor $2^{O(d)}$ to $O(1)$. At each iteration, each vector output by the sieve is a subtraction of two input vectors. In other words, the sieve will select a subset C of the initial set S, and the output set will be obtained by subtracting a vector of C to each vector of $S \setminus C$. By volume arguments, one can choose a set C, which is never too large, so that the number of samples does not decrease too much. Intuitively, one uses the fact that for any $0 < \gamma < 1$, a ball of radius R can be recovered by at most an exponential number of balls of radius γR.

We just described the principles of the AKS algorithm [14], but the proved algorithm is a bit more complex, and its analysis is nontrivial.

HKZ Reduction

It is easy to see that any exact SVP algorithm allows to find an HKZ-reduced basis, within the same asymptotic running time, by calling the algorithm a linear number of times. For instance, one can do as follows:

- Call the SVP algorithm on L to obtain a shortest vector $\mathbf{b}_1$ of the lattice L.
- Extend $\mathbf{b}_1$ into a basis $(\mathbf{b}_1, \mathbf{c}_2, \ldots, \mathbf{c}_d)$ of L and compute a basis of the projected lattice $\pi_2(L)$.
- Call the SVP algorithm on $\pi_2(L)$ to obtain a shortest vector $\mathbf{b}_2'$ of the projected lattice $\pi_2(L)$.
- Lift $\mathbf{b}_2'$ into a vector $\mathbf{b}_2$ of L by adding an appropriate multiple of $\mathbf{b}_1$ so that $(\mathbf{b}_1, \mathbf{b}_2)$ is size-reduced.
- Extend $(\mathbf{b}_1, \mathbf{b}_2)$ into a basis $(\mathbf{b}_1, , \mathbf{b}_2, \mathbf{c}_3, \ldots, \mathbf{c}_d)$ of L and use this basis to compute a basis of the projected lattice $\pi_3(L)$. And so on.

Mordell's Inequality and Blockwise Algorithms

We saw in section "Hermite's Inequality and the Lenstra–Lenstra–Lovász Algorithm" the LLL algorithm [16] (see Corollary 4): given a basis of an d-dimensional integer lattice $L \subseteq \mathbb{Z}^n$ and a reduction factor $\varepsilon > 0$, LLL outputs (in time polynomial in $1/\varepsilon$ and the size of the basis) a reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ whose first vector is provably short, namely,

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \le \left((1+\varepsilon)\sqrt{4/3}\right)^{(d-1)/2}, \tag{2.28}$$

$$\|\mathbf{b}_1\|/\lambda_1(L) \le \left((1+\varepsilon)\sqrt{4/3}\right)^{d-1}. \tag{2.29}$$

We noted that the first inequality (2.28) was reminiscent of Hermite's inequality [7] on γ_d:

$$\gamma_d \le \left(\sqrt{4/3}\right)^{d-1} = \gamma_2^{d-1}, \quad \text{(Hermite's inequality)} \tag{2.30}$$

which means that L has a nonzero vector of norm $\leq (\sqrt{4/3})^{(d-1)/2}\mathrm{vol}(L)^{1/d}$. Thus, we viewed LLL as an algorithmic version of Hermite's inequality (2.21), and this connection was strengthened by the fact that LLL is a variant of an algorithm introduced by Hermite [7] to prove (2.21), based on Lagrange's two-dimensional algorithm [5].

The second inequality (2.29) means that LLL approximates the shortest vector problem (SVP) within an exponential factor. On the other hand, we saw in section "Solving Exact SVP" the best algorithms for exact-SVP, which are exponential: Kannan's deterministic algorithm [10] requires $2^{O(d\,\log d)}$ polynomial-time operations, and the AKS probabilistic algorithm [14] requires $2^{O(d)}$ polynomial-time operations.

A natural question is whether the upper bounds of (2.28) or (2.29) can be decreased in polynomial time. The only polynomial-time algorithms achieving better inequalities than (2.28) or (2.29) are blockwise generalizations of LLL: Schnorr's algorithm [17], the transference algorithm by Gama et al. [18], and Gama–Nguyen's slide algorithm [19], the latter one offering better theoretical guarantees than the first two. Blockwise algorithms rely on a SVP-subroutine [10, 14] (see section "Solving Exact SVP") computing shortest vectors in smaller lattices of dimension $\leq k$, where k is an additional input parameter referred to as the blocksize. Note that the exponential cost of the SVP-subroutine can be kept polynomial in the size of the basis if the blocksize k is sufficiently small: namely, $k = O(\log d)$ (resp. $k = O(\log d/\log\log d)$) suffices with AKS [14] (respectively [10]) as the SVP subroutine. As the cost of the SVP-subroutine is exponential in the blocksize, it is important to use the SVP-subroutine as efficiently as possible for a given output quality.

In this section, we will describe Gama–Nguyen's slide algorithm [19], which improves [17, 18], and is simpler in several respects. For instance, it might be argued that the inequalities achieved by [17, 18] are not very natural: more precisely, in Schnorr's algorithm [17], k must be even, d must be a multiple of $k/2$, and the upper bound of (2.29) is replaced by $\sqrt{2\gamma_{k/2}\alpha_{k/2}}((1+\varepsilon)\beta_{k/2})^{d/k-1}$, where $\alpha_{k/2}$ and $\beta_{k/2}$ are technical constants bounded in [13, 17, 18]; and in the GHKN algorithm [18], the upper bound of (2.28) is replaced by $\gamma_{k-1}^{(d+k-1)/(4(k-1))}((1+\varepsilon)\gamma_k)^{k(d-k+1)/(4(k-1)^2)}$, while the upper bound of (2.29) is replaced by the square of the previous expression. The new algorithm [19] is a blockwise algorithm achieving better and more "natural" upper bounds, corresponding to the following classical generalization of Hermite's inequality (2.21), known as Mordell's inequality [20,48]:

Theorem 11 (Mordell's Inequality [48]). *For all integers d and k such that $2 \leq k \leq d$:*

$$\gamma_d \leq \gamma_k^{(d-1)/(k-1)} \tag{2.31}$$

This implies that any d-rank lattice L has a nonzero vector of norm:

$$\leq \sqrt{\gamma_k}^{(d-1)/(k-1)} \mathrm{vol}(L)^{1/d}.$$

By analogy with the LLL case, Mordell's inequality (2.31) suggests that there might exist a blockwise reduction algorithm calling polynomially many times a SVP-subroutine in dimension $\leq k$, and which outputs a basis whose first vector $\mathbf{b}_1 \in L$ would satisfy

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \leq \sqrt{(1+\varepsilon)\gamma_k}^{(d-1)/(k-1)} \tag{2.32}$$

Such an algorithm would be a polynomial-time version of Mordell's inequality, just as LLL is a polynomial-time version of Hermite's inequality. And an old result of Lovász [32] shows that by calling d times such an algorithm, we would also obtain a nonzero lattice vector $\mathbf{b}_1 \in L$ satisfying

$$\|\mathbf{b}_1\|/\lambda_1(L) \leq ((1+\varepsilon)\gamma_k)^{(d-1)/(k-1)} \tag{2.33}$$

Note that (2.28) and (2.29) are exactly the $k = 2$ case of (2.32) and (2.33). Unfortunately, the classical proof [20] of Mordell's inequality (2.31) does not give such an algorithm. And the blockwise algorithms [17, 18] turn out to be loose algorithmic versions of Mordell's inequality: for any k, the best upper bounds known on $\|\mathbf{b}_1\|$ for [17, 18] are worse than (2.32) and (2.33). For instance, the best upper bound known on $\|\mathbf{b}_1\|/\lambda_1(L)$ for Schnorr's algorithm is essentially $\left((1+\varepsilon)(k/2)^{2\ln 2}\right)^{d/k-1}$.

Slide reduction [19] is an algorithmic version of Mordell's inequality in the following sense: given a basis of an d-dimensional integer lattice $L \subseteq \mathbb{Z}^n$, a blocksize k dividing d, a reduction factor $\varepsilon > 0$, and a SVP-subroutine computing shortest vectors in any lattice of dimension $\leq k$, slide reduction outputs (in time polynomial in the size of the basis and $1/\varepsilon$) a basis whose first vector $\mathbf{b}_1$ satisfies (2.32) and the following inequality:

$$\|\mathbf{b}_1\|/\lambda_1(L) \leq ((1+\varepsilon)\gamma_k)^{(d-k)/(k-1)}, \tag{2.34}$$

and the number of calls to the SVP-subroutine is polynomial in the size of the basis and $1/\varepsilon$. Surprisingly, (2.34) is slightly better than the speculated inequality (2.33), by a multiplicative factor close to γ_k. Hence, slide reduction is theoretically better than Schnorr's algorithm [17] and Gama *et al.*'s transference algorithm [18] for any fixed k, but does not improve the asymptotical sub-exponential approximation factor when $k = O(\log d)$.

Like all known proofs of Mordell's inequality, slide reduction is based on duality. Furthermore, it was proved in [19] that in the worst case, (2.32) and (2.34) are essentially tight: namely, there exist slide reduced bases such that these upper bounds become lower bounds if we replace γ_k by a slightly smaller linear function of k, namely $\gamma_k/2$ or even $(1-\varepsilon')k/(2\pi e)$ for all $\varepsilon' > 0$. Ajtai proved [49] an analogue result for Schnorr's algorithm [17], without effective constants.

Classical Proofs of Mordell's Inequality

We give here the classical argument showing Mordell's inequality (2.31), such as the one given in [20, Theorem 2.3.1]: this argument can actually be found earlier than Mordell's article [48], for instance when Korkine and Zolotarev [28] determined the value of γ_4 by showing first that $\gamma_4 \leq \gamma_3^{3/2}$, and also somewhat implicitly in Hermite's first letter [7].

We first notice that it suffices to show the inequality for $k = d - 1$: indeed, if (2.31) holds for $k = d - 1$, then by applying recursively the inequality, we obtain (2.31) for all k. In fact, Mordell's inequality is equivalent to showing that the sequence $(\gamma_d^{1/(d-1)})_{d\geq 2}$ decreases.

Let L be a d-rank lattice. Let $\mathbf{x}$ be a shortest nonzero vector of the dual lattice $L^\times$ and let H be the hyperplane $\mathbf{x}^\perp$. Denote by M the $(d-1)$-rank lattice $L \cap H$. Then $\mathrm{vol}(M) = \mathrm{vol}(L)\|\mathbf{x}\|$ and $\|\mathbf{x}\| \leq \sqrt{\gamma_d}\mathrm{vol}(L^\times)^{1/d} = \sqrt{\gamma_d}\mathrm{vol}(L)^{-1/d}$; therefore,

$$\mathrm{vol}(M) \leq \sqrt{\gamma_d}\mathrm{vol}(L)^{1-1/d}.$$

In particular,

$$\lambda_1(M) \leq \sqrt{\gamma_{d-1}}\left(\sqrt{\gamma_d}\mathrm{vol}(L)^{1-1/d}\right)^{1/(d-1)} = \sqrt{\gamma_{d-1}}\sqrt{\gamma_d}^{1/(d-1)}\mathrm{vol}(L)^{1/d}.$$

Furthermore, we have $\lambda_1(L) \leq \lambda_1(M)$. Hence, by definition of γ_d,

$$\sqrt{\gamma_d} \leq \sqrt{\gamma_{d-1}}\sqrt{\gamma_d}^{1/(d-1)}.$$

The proof of (2.31) is now over, since we can rewrite the previous inequality as

$$\gamma_d \leq \gamma_{d-1}^{(d-1)/(d-2)}.$$

This classical proof of Mordell's inequality cannot be directly translated into a recursive algorithm: indeed, it considers shortest vectors in the $(d-1)$-rank lattice M, and also in the d-rank lattice $L^\times$. In the next subsection, we slightly modify the argument so that only $(d-1)$-rank lattices are considered, which naturally gives rise to algorithms.

Mordell's Inequality by Reduction

We introduce the following reduction notion, which we dub Mordell's reduction because it is inspired by Mordell's inequality or rather its proof:

Definition 18. Let $d \geq 2$. A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L is *Mordell-reduced* with factor $\varepsilon \geq 0$ if and only if the following two conditions hold:

$$\|\mathbf{b}_1\| = \lambda_1(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_{d-1})) \tag{2.35}$$

and

$$1/\|\mathbf{b}_d^\star\| \leq (1+\varepsilon)\lambda_1(\pi_2(L)^\times), \tag{2.36}$$

where $\pi_2(L)$ denotes the orthogonal projection of L over the hyperplane $\mathbf{b}_1^\perp$, and $\mathbf{b}_d^\star$ denotes as usual the component of $\mathbf{b}_d$ which is orthogonal to the hyperplane spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_{d-1}$.

The inequality (2.36) is motivated by the fact that $\mathbf{b}_d^\star/\|\mathbf{b}_d^\star\|^2 \in \pi_2(L)^\times$ (which we previously mentioned at the end of section "Gram–Schmidt and Triangularization" giving a link between duality and Gram–Schmidt orthogonalization), because the vector is orthogonal with $\mathbf{b}_1, \ldots, \mathbf{b}_{d-1}$, and its dot product with $\mathbf{b}_d$ is equal to 1.

Note that there always exist Mordell-reduced bases for all $\varepsilon \geq 0$. Indeed, consider an HKZ-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L. Then (2.35) holds. Next, consider a shortest vector $\mathbf{c}$ in $\pi_2(L)^\times$ and modify $\mathbf{b}_2, \ldots, \mathbf{b}_d$ in such a way that $\mathbf{b}_d^\star/\|\mathbf{b}_d^\star\| = \mathbf{c}$ and $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ remains a basis of L: then both (2.36) and (2.35) hold.

Mordell's reduction has the following properties:

Lemma 10. *Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be a Mordell-reduced basis of L with factor $\varepsilon \geq 0$ and $d \geq 3$. Then*

1. *Primal inequality:*

$$\|\mathbf{b}_1\| \leq \sqrt{\gamma_{d-1}}^{(d-1)/(d-2)} \left(\prod_{i=2}^{d-1} \|\mathbf{b}_i^\star\|\right)^{1/(d-2)}. \tag{2.37}$$

2. *Dual inequality:*

$$\left(\prod_{i=2}^{d-1} \|\mathbf{b}_i^\star\|\right)^{1/(d-2)} \leq \left((1+\varepsilon)\sqrt{\gamma_{d-1}}\right)^{(d-1)/(d-2)} \|\mathbf{b}_d^\star\|. \tag{2.38}$$

3. *Primal–dual inequality:*

$$\|\mathbf{b}_1^\star\|/\|\mathbf{b}_d^\star\| \leq ((1+\varepsilon)\gamma_{d-1})^{(d-1)/(d-2)}. \tag{2.39}$$

4. *Relaxed Mordell's inequality:*

$$\|\mathbf{b}_1\| \leq \left((1+\varepsilon)^{1/d}\sqrt{\gamma_{d-1}}\right)^{(d-1)/(d-2)} \mathrm{vol}(L)^{1/d}. \tag{2.40}$$

Proof. Equation (2.37) follows from $\|\mathbf{b}_1\| = \lambda_1(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_d))$ and the definition of γ_d. Indeed, we have

$$\|\mathbf{b}_1\| \leq \sqrt{\gamma_{d-1}} \left(\prod_{i=1}^{d-1} \|\mathbf{b}_i^\star\| \right)^{1/(d-1)} .$$

Therefore,

$$\|\mathbf{b}_1\|^{d-1} \leq \sqrt{\gamma_{d-1}}^{d-1} \prod_{i=1}^{d-1} \|\mathbf{b}_i^\star\|,$$

which can be rewritten as (2.37). Similarly, $1/\|\mathbf{b}_d^\star\| \leq (1+\varepsilon)\lambda_1(\pi_2(L)^\times)$ implies that

$$1/\|\mathbf{b}_d^\star\| \leq (1+\varepsilon)\sqrt{\gamma_{d-1}} \left(\prod_{i=2}^{d} 1/\|\mathbf{b}_i^\star\| \right)^{1/(d-1)} ;$$

therefore,

$$\prod_{i=2}^{d} \|\mathbf{b}_i^\star\| \leq \left((1+\varepsilon)\sqrt{\gamma_{d-1}}/\|\mathbf{b}_d^\star\|\right)^{d-1},$$

which implies (2.38). And (2.39) follows from multiplying (2.37) and (2.38). Furthermore, we have

$$\begin{aligned}
\mathrm{vol}(L) &= \prod_{i=1}^{d} \|\mathbf{b}_i^\star\| \\
&= \|\mathbf{b}_d^\star\| \times \|\mathbf{b}_1^\star\| \times \prod_{i=2}^{d-1} \|\mathbf{b}_i^\star\| \\
&\geq \frac{\left(\prod_{i=2}^{d-1} \|\mathbf{b}_i^\star\|\right)^{1/(d-2)}}{((1+\varepsilon)\sqrt{\gamma_{d-1}})^{(d-1)/(d-2)}} \times \|\mathbf{b}_1^\star\| \times \prod_{i=2}^{d-1} \|\mathbf{b}_i^\star\| \quad \text{by (2.38)} \\
&= \frac{\|\mathbf{b}_1^\star\|}{((1+\varepsilon)\sqrt{\gamma_{d-1}})^{(d-1)/(d-2)}} \times \left(\prod_{i=2}^{d-1} \|\mathbf{b}_i^\star\| \right)^{1+1/(d-2)} \\
&\geq \frac{\|\mathbf{b}_1^\star\|}{((1+\varepsilon)\sqrt{\gamma_{d-1}})^{(d-1)/(d-2)}} \times \left(\frac{\|\mathbf{b}_1^\star\|}{\sqrt{\gamma_{d-1}}^{(d-1)/(d-2)}} \right)^{(d-2)+1} \quad \text{by (2.37)} \\
&= \frac{\|\mathbf{b}_1^\star\|^d}{(1+\varepsilon)^{(d-1)/(d-2)} \sqrt{\gamma_{d-1}}^{(1+(d-2)+1)(d-1)/(d-2)}} \\
&= \frac{\|\mathbf{b}_1^\star\|^d}{(1+\varepsilon)^{(d-1)/(d-2)} \sqrt{\gamma_{d-1}}^{d(d-1)/(d-2)}},
\end{aligned}$$

which proves (2.40). □

Theorem 12. *Let $k \geq 2$. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a basis of a lattice L such that $(\mathbf{b}_1, \ldots, \mathbf{b}_{k+1})$ is Mordell-reduced and $\mathbf{b}_{k+1}^\star$ is a shortest vector in the projected* *lattice* $\pi_{k+1}(L)$. *Then*

$$\frac{\prod_{i=1}^{k}\|\mathbf{b}_i^\star\|}{\prod_{i=k+1}^{2k}\|\mathbf{b}_i^\star\|} \leq ((1+\varepsilon)\gamma_k)^{k^2/(k-1)}\,. \tag{2.41}$$

Proof. As $\mathbf{b}_{k+1}^\star$ is a shortest vector of the projected lattice $\pi_{k+1}(L)$, we can apply (2.37) to obtain

$$\|\mathbf{b}_{k+1}^\star\| \leq \sqrt{\gamma_k}^{k/(k-1)}\left(\prod_{i=k+2}^{2k}\|\mathbf{b}_i^\star\|\right)^{1/(k-1)} \quad ;$$

therefore, we can lower bound the denominator of (2.41) as

$$\prod_{i=k+1}^{2k}\|\mathbf{b}_i^\star\| \geq \|\mathbf{b}_{k+1}^\star\| \times \left(\frac{\|\mathbf{b}_{k+1}^\star\|}{\sqrt{\gamma_k}^{k/(k-1)}}\right)^{k-1} = \|\mathbf{b}_{k+1}^\star\|^k/\sqrt{\gamma_k}^k. \tag{2.42}$$

On the other hand, $(\mathbf{b}_1,\ldots,\mathbf{b}_{k+1})$ is Mordell-reduced, so (2.38) implies that

$$\prod_{i=2}^{k}\|\mathbf{b}_i^\star\| \leq \left((1+\varepsilon)\sqrt{\gamma_k}\right)^k\|\mathbf{b}_{k+1}^\star\|^{k-1},$$

and (2.39) implies that

$$\|\mathbf{b}_1^\star\| \leq ((1+\varepsilon)\gamma_k)^{k/(k-1)} \times \|\mathbf{b}_{k+1}^\star\|.$$

By multiplying the previous two inequalities, we can upper bound the numerator of (2.41) as

$$\prod_{i=1}^{k}\|\mathbf{b}_i^\star\| \leq \|\mathbf{b}_{k+1}^\star\|^k \times ((1+\varepsilon)\gamma_k)^{k/(k-1)} \times \left((1+\varepsilon)\sqrt{\gamma_k}\right)^k. \tag{2.43}$$

Hence, (2.43) and (2.42) imply that

$$\begin{aligned}\frac{\prod_{i=1}^{k}\|\mathbf{b}_i^\star\|}{\prod_{i=k+1}^{2k}\|\mathbf{b}_i^\star\|} &\leq ((1+\varepsilon)\gamma_k)^{k/(k-1)} \times \left((1+\varepsilon)\sqrt{\gamma_k}\right)^k \times \sqrt{\gamma_k}^k\\ &= ((1+\varepsilon)\gamma_k)^{k+k/(k-1)}\\ &= ((1+\varepsilon)\gamma_k)^{k^2/(k-1)},\end{aligned}$$

which proves (2.41). □

We later show (and it is not difficult to see) that there exist bases satisfying the assumptions of Theorem 12 for any $\varepsilon \geq 0$: by taking $\varepsilon = 0$, this proves that for all $k \geq 2$

$$\gamma_{2k,k} \leq \gamma_k^{k^2/(k-1)}.$$

Blockwise Reduction

For any basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_d]$, we use the notation $B_{[i,j]}$ for the projected block $[\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_j)]$, where π_i is the orthogonal projection over $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^{\perp}$. When looking at the lower-triangular representation of B, $B_{[i,j]}$ corresponds to the (lower-triangular) submatrix of the lower-triangular matrix within row i to row j. Note that $B_{[i,j]}$ always represents a linearly independent family of $j - i + 1$ vectors, whose first vector is $\mathbf{b}_i^{\star}$. For example, $B_{[i,i]} = [\mathbf{b}_i^{\star}]$ and $B_{[1,i]} = [\mathbf{b}_1, \ldots, \mathbf{b}_i]$ for all $i \in [1, d]$. If B has integer coefficients, then $B_{[i,j]}$ has rational coefficients if $i > 1$ and integer coefficients if $i = 1$. As an important particular case, if T is a lower triangular matrix (such as the μ matrix of the Gram–Schmidt orthogonalization), then $T_{[i,j]}$ is simply the inner triangular matrix within the indices $[i, j]$.

In the LLL algorithm, vectors are considered two by two. At each loop iteration, the two-dimensional lattice $L_i = [\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1})]$ is partially reduced (through a swap) to decrease $\|\mathbf{b}_i^{\star}\|$ by at least some geometric factor. When all such lattices are almost reduced, every ratio $\|\mathbf{b}_i^{\star}\|/\|\mathbf{b}_{i+1}^{\star}\|$ is roughly less than $\gamma_2 = \sqrt{\frac{4}{3}}$.

In blockwise generalizations of LLL, we select an integer $k \geq 2$ dividing d, called the blocksize. Then, the vectors $\mathbf{b}_i^{\star}$ are "replaced" by k-dimensional blocks $S_i = B_{[ik-k+1,ik]}$, where $1 \leq i \leq \frac{d}{k}$. The analogue of the two-dimensional L_i in LLL are the $2k$-dimensional large blocks $L_i = B_{[ik-k+1,ik+k]}$, where $1 \leq i \leq \frac{d}{k} - 1$. The link between the small blocks $S_1, \ldots, S_{d/k}$ and the large blocks $L_1, \ldots, L_{d/k-1}$ is that S_i consists of the first k vectors of L_i, while S_{i+1} is the projection of the last k vectors of L_i over $\mathrm{span}(S_i)^{\perp}$. As a result, $\mathrm{vol}(L_i) = \mathrm{vol}(S_i) \times \mathrm{vol}(S_{i+1})$. By analogy with LLL, the blockwise algorithm will perform operations on each large block L_i so that $\mathrm{vol}(S_i)/\mathrm{vol}(S_{i+1})$ can be upper bounded.

Gama and Nguyen [19] introduced the following blockwise version of Mordell's reduction (in fact, the reduction in [19] is a bit stronger, but the difference is minor and not relevant):

Definition 19. Let $d \geq 2$ and $k \geq 2$ dividing d. A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L is *block-Mordell-reduced* with factor $\varepsilon \geq 0$ and blocksize k if and only if it is size-reduced and the following two conditions hold:

- For each $i \in \{1, \ldots, d/k - 1\}$, the block $B_{[ik-k+1,ik+1]}$ is Mordell-reduced.
- We have, $\|\mathbf{b}_{d-k+1}^{\star}\| = \lambda_1(\mathcal{L}(B_{[d-k+1,d]}))$.

This is equivalent to asking that the basis is size-reduced and the following two conditions hold:

1. *Primal conditions*: for each $j \in \{1, \ldots, d\}$ such that $j \equiv 1 \pmod{k}$,

$$\|\mathbf{b}_j^\star\| = \lambda_1(\mathcal{L}(B_{[j,j+k-1]})). \tag{2.44}$$

   Note that $B_{[j,j+k-1]}$ is one of the small blocks S_i, namely $S_{1+(j-1)/k}$.

2. *Dual conditions*: for each $j \in \{1, \ldots, d-k\}$ such that $j \equiv 1 \pmod{k}$,

$$1/\|\mathbf{b}_{j+k}^\star\| \leq (1+\varepsilon)\lambda_1(\mathcal{L}(B_{[j+1,j+k]})^\times). \tag{2.45}$$

   Note that $B_{[j+1,j+k]}$ is not one of the small blocks S_i, because there is a shift of index: the block starts at index $j+1$ rather than j.

Let us explain the intuition behind block-Mordell reduction. Conditions (2.44) and (2.45) imply that each vector $\mathbf{b}_j^\star$ such that $j \in \{k, \ldots, d\}$ and $j \equiv 1 \pmod{k}$ is neither too large, nor too short:

- Not too large because $\|\mathbf{b}_j^\star\| = \lambda_1(\mathcal{L}(B_{[j,j+k-1]}))$;
- Not too short because $1/\|\mathbf{b}_j^\star\| \leq (1+\varepsilon)\lambda_1(\mathcal{L}(B_{[j-k+1,j]})^\times)$.

These conditions are inspired by the fact that $\mathbf{b}_j^\star$ is connected to two natural k-rank lattices:

- $\mathbf{b}_j^\star$ belongs to the projected lattice $\mathcal{L}(B_{[j,j+k-1]})$: it is in fact the first vector of $B_{[j,j+k-1]}$.
- $\mathbf{b}_j^\star/\|\mathbf{b}_j^\star\|^2$ belongs to the dual-projected lattice $\mathcal{L}(B_{[j-k+1,j]})^\times$: see the end of section "Gram–Schmidt and Triangularization" for links between duality and Gram–Schmidt orthogonalization.

We now give elementary properties of block-Mordell-reduced bases, which follow from Mordell reduction:

Lemma 11. *Let* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ *be a block-Mordell-reduced basis of a lattice* L *with factor* $\varepsilon \geq 0$ *and blocksize* $k \geq 2$ *dividing* d*. Then,*

1. *Primal inequality: for each* $j \in \{1, \ldots, d\}$ *such that* $j \equiv 1 \pmod{k}$,

$$\|\mathbf{b}_j^\star\| \leq \sqrt{\gamma_k}^{k/(k-1)} \left(\prod_{i=j+1}^{j+k-1} \|\mathbf{b}_i^\star\| \right)^{1/(k-1)}. \tag{2.46}$$

2. *Dual inequality: for each* $j \in \{1, \ldots, d-k\}$ *such that* $j \equiv 1 \pmod{k}$,

$$\left(\prod_{i=j+1}^{j+k-1} \|\mathbf{b}_i^\star\| \right)^{1/(k-1)} \leq \left((1+\varepsilon)\sqrt{\gamma_k}\right)^{k/(k-1)} \|\mathbf{b}_{j+k}^\star\|. \tag{2.47}$$

3. *Primal-dual inequality: for each* $j \in \{1, \ldots, d-k\}$ *such that* $j \equiv 1 \pmod{k}$,

$$\|\mathbf{b}_j^\star\|/\|\mathbf{b}_{j+k}^\star\| \leq ((1+\varepsilon)\gamma_k)^{k/(k-1)}. \tag{2.48}$$

4. Half-volume inequality: for each $j \in \{1, \ldots, d-k\}$ *such that* $j \equiv 1 \pmod{k}$,

$$\frac{\prod_{i=j}^{j+k-1} \|\mathbf{b}_i^\star\|}{\prod_{i=j+k}^{j+2k-1} \|\mathbf{b}_i^\star\|} \leq ((1+\varepsilon)\gamma_k)^{k^2/(k-1)}. \tag{2.49}$$

Proof. Equation (2.46) follows from (2.37), (2.47) follows from (2.38), (2.48) follows from (2.39), and (2.49) follows from (2.41). □

Theorem 13. *Let* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ *be a block-Mordell-reduced basis of a lattice L with factor* $\varepsilon \geq 0$ *and blocksize* $k \geq 2$ *dividing d. Then,*

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \leq \sqrt{\gamma_k}^{(d-1)/(k-1)} \times \sqrt{1+\varepsilon}^{(d-k)/(k-1)}. \tag{2.50}$$

$$\|\mathbf{b}_1\|/\lambda_1(L) \leq ((1+\varepsilon)\gamma_k)^{(d-k)/(k-1)}, \tag{2.51}$$

Proof. We have,

$$\mathrm{vol}(L) = \prod_{i=1}^{d/k} \mathrm{vol}(S_i),$$

where, by (2.49), for each $i \in \{1, \ldots, d/k-1\}$:

$$\mathrm{vol}(S_i)/\mathrm{vol}(S_{i+1}) \leq ((1+\varepsilon)\gamma_k)^{k^2/(k-1)}.$$

This implies that, similar to LLL,

$$\mathrm{vol}(S_1) \leq ((1+\varepsilon)\gamma_k)^{k^2/(k-1)\times(d/k-1)/2} \mathrm{vol}(L)^{1/(d/k)}.$$

And (2.44) implies that $\|\mathbf{b}_1^\star\| = \lambda_1(\mathcal{L}(B_{[1,k]})) = \lambda_1(S_1)$; therefore,

$$\begin{aligned}
\|\mathbf{b}_1^\star\| &\leq \sqrt{\gamma_k}\mathrm{vol}(S_1)^{1/k} \\
&\leq \sqrt{\gamma_k}\,((1+\varepsilon)\gamma_k)^{k/(k-1)\times(d/k-1)/2} \mathrm{vol}(L)^{1/d} \\
&= \sqrt{\gamma_k}^{1+(d-k)/(k-1)}(1+\varepsilon)^{(d-k)/(2(k-1))}\mathrm{vol}(L)^{1/d} \\
&= \sqrt{\gamma_k}^{(d-1)/(k-1)}(1+\varepsilon)^{(d-k)/(2(k-1))}\mathrm{vol}(L)^{1/d},
\end{aligned}$$

which implies (2.50). Now, consider a shortest vector $\mathbf{u}$ of L. Then $\|\mathbf{u}\| = \lambda_1(L)$ and $\mathbf{u}$ can be written as $\mathbf{u} = \sum_{i=1}^m \alpha_i \mathbf{b}_i$, where each $\alpha_i \in \mathbb{Z}$ and $\alpha_m \neq 0$. If we let $q = \lfloor (m-1)/k \rfloor$, then $\pi_{qk+1}(\mathbf{u})$ is a nonzero vector of $L(B_{[qk+1,qk+k]})$. But by definition of block-Mordell reduction, $\mathbf{b}_{qk+1}^\star$ is a shortest vector of $L(B_{[qk+1,qk+k]})$; therefore,

$$\|\mathbf{b}_{qk+1}^\star\| \leq \|\pi_{qk+1}(\mathbf{u})\| \leq \|\mathbf{u}\| = \lambda_1(L),$$

which implies that

$$\|\mathbf{b}_1\|/\lambda_1(L) \leq \|\mathbf{b}_1\|/\|\mathbf{b}^\star_{qk+1}\|.$$

However, note that

$$\frac{\|\mathbf{b}_1\|}{\|\mathbf{b}^\star_{qk+1}\|} = \prod_{i=0}^{q-1} \frac{\|\mathbf{b}^\star_{ik+1}\|}{\|\mathbf{b}^\star_{(i+1)k+1}\|},$$

which, by (2.48), is

$$\leq \left(((1+\varepsilon)\gamma_k)^{k/(k-1)}\right)^q = ((1+\varepsilon)\gamma_k)^{qk/(k-1)},$$

where $qk \leq d-k$. Hence,

$$\|\mathbf{b}_1\|/\lambda_1(L) \leq ((1+\varepsilon)\gamma_k)^{(d-k)/(k-1)},$$

which proves (2.51). □

The Slide Algorithm

Gama and Nguyen [19] presented a polynomial-time algorithm to block-Mordell-reduce a basis, using an SVP-oracle in dimension $\leq k$: Algorithm 7 is a simplified version, to make exposition easier. By an SVP-oracle, [19] means any algorithm which, given as input the Gram matrix of a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ of an integer lattice L, outputs $(u_1, \ldots, u_k) \in \mathbb{Z}^k$ such that $\| \sum_{i=1}^k u_i \mathbf{b}_i \| = \lambda_1(L)$.

```
Input: a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice $L$, together with a reduction factor $\varepsilon \geq 0$ and a blocksize
    $k \geq 2$ dividing $d$.
Output: the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is block-Mordell-reduced with factor $\varepsilon$ and blocksize $k \geq 2$.
1: LLL-reduce $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ using Algorithm 6.
2: if there exists $j \in \{1, \ldots, d\}$ such that $j \equiv 1 \pmod{k}$ and $j$ does not satisfy (2.44) then
3:     Use an SVP-oracle in dimension $\leq k$ to locally HKZ-reduce the block $B_{[j,j+k-1]}))$, which
       implies that (2.44) holds; then return to Step 1. Basis vectors outside the block $B_{[j,j+k-1]}))$
       are not modified.
4: end if
5: if there exists $j \in \{1, \ldots, d-k\}$ such that $j \equiv 1 \pmod{k}$ and $j$ does not satisfy (2.45)
   then
6:     Use an SVP-oracle in dimension $\leq k$ to reduce the block $B_{[j+1,j+k]}$ in such a way that
       $1/\|\mathbf{b}^\star_{j+k}\| = \lambda_1(\mathcal{L}(B_{[j+1,j+k]})^\times)$, which implies that (2.45) holds; then return to Step 1.
       Basis vectors outside the block $B_{[j+1,j+k]}$ are not modified.
7: end if
```

Algorithm 7: The basic slide algorithm [19]

Tests in Steps 2 and 5 are performed using an SVP-oracle in dimension k. We will not describe the local reductions performed in Steps 3 and 6: they are natural and are presented in [19]. Their cost is a linear number of calls to an SVP-oracle in dimension $\leq k$, together with polynomial-time operations, like an HKZ-reduction of a k-dimensional basis.

What is clear is that if the slide algorithm of Fig. 7 terminates, then the final basis is block-Mordell-reduced with factor ε and blocksize k. What is less clear is why the algorithm terminates, and what is its complexity. By analogy with the complexity analysis of LLL, one considers the following integral potential:

$$D' = \prod_{i=1}^{d/k} \text{vol}(\mathcal{L}(B_{[1,ik]}))^2 \in \mathbb{Z}^+.$$

Then D can be rewritten as

$$D' = \prod_{i=1}^{d/k} \prod_{j=1}^{i} \text{vol}(S_j)^2 = \prod_{j=1}^{d/k} \text{vol}(S_j)^{2(d/k+1-j)}, \tag{2.52}$$

which is the blockwise analogue of $D = \|\mathbf{b}_1^\star\|^{2d} \|\mathbf{b}_2^\star\|^{2(d-1)} \times \cdots \times \|\mathbf{b}_d^\star\|^2$, which was used for analyzing LLL. Clearly, log D' is initially polynomial in the size of the basis.

We use D' to show that the number of times that the slide algorithm (Algorithm 7) goes through Step 1 is polynomially bounded, just as D was used to show that number of swaps in LLL was polynomially bounded. Let us look at the operations of Algorithm 7, which could possibly modify the integer D': it turns out that only Steps 1 and 6 can modify D', because Step 3 only modifies one block S_i (for some i), but the volume of this block cannot change, as the volume of the whole lattice remains the same. We discuss Steps 1 and 6 separately:

- Step 1 is an LLL reduction, which performs only size-reductions and swaps. Size-reductions do not modify any of the $\mathbf{b}_i^\star$, and therefore cannot modify D'. And we note that swaps of vectors $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ can modify D' only if $i \equiv 1 \pmod{k}$. When this is the case, $i = 1 + k\ell$ for some integer $\ell \geq 1$, and we see that the last vector of the block $S_{\ell-1}$ is the projection of $\mathbf{b}_{i-1}$, while the first vector of the block S_ℓ is the projection of $\mathbf{b}_i$. This means that in (2.52) of D', only vol$(S_{\ell-1})$ and vol(S_ℓ) may change. On the other hand, vol$(S_{\ell-1}) \times vol(S_\ell)$ remains the same because vol$(L) = \prod_{j=1}^{d/k}$ vol(S_j) cannot change. But if LLL swapped $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$, this means that Lovász' condition failed for $(i-1, i)$, which implies that $\|\mathbf{b}_{i-1}^\star\|$ will decrease strictly (in fact, by some multiplicative factor < 1): in this case, vol$(S_{\ell-1})$ will decrease, and therefore D'. Hence, only two situations can occur:

Case 1: Step 1 never swaps vectors $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ such that $i \equiv 1 \pmod{k}$, in which case D' does not change. Here, the swaps are always within a block S_ℓ, never between two consecutive blocks $S_{\ell-1}$ and S_ℓ.

Case 2: Step 1 swaps at least once a pair of vectors $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ such that $i \equiv 1 \pmod{k}$, in which case D' decreases by some multiplicative factor < 1 depending on ε. This means that this situation occurs at most polynomially many times.

- Step 6 modifies the block $B_{[j+1,j+k]}$ so that $1/\|\mathbf{b}^\star_{j+k}\| = \lambda_1(\mathcal{L}(B_{[j+1,j+k]})^\times)$, which implies (2.45). As $j \equiv 1 \pmod{k}$, we may write $j = 1 + k\ell$ for some integer $\ell \geq 0$. We see that in (2.52) of D', only $\mathrm{vol}(S_{\ell+1})$ and $\mathrm{vol}(S_{\ell+2})$ change. On the other hand, $\mathrm{vol}(S_{\ell+1}) \times vol(S_{\ell+2})$ remains the same because $\mathrm{vol}(L) = \prod_{j=1}^{d/k} \mathrm{vol}(S_j)$ cannot change. Before Step 6, (2.45) did not hold, which means that $1/\|\mathbf{b}^\star_{j+k}\| > (1+\varepsilon)\lambda_1(\mathcal{L}(B_{[j+1,j+k]})^\times)$. But after Step 6, we have $1/\|\mathbf{b}^\star_{j+k}\| = \lambda_1(\mathcal{L}(B_{[j+1,j+k]})^\times)$, which implies that $1/\|\mathbf{b}^\star_{j+k}\|$ decreases by a multiplicative factor $\leq 1/(1+\varepsilon) < 1$. As $\mathbf{b}^\star_{j+k}$ is the first vector of $S_{\ell+2}$, this means that $\mathrm{vol}(S_{\ell+2})$ increases by a multiplicative factor $\geq 1+\varepsilon$, and therefore $\mathrm{vol}(S_{\ell+1})$ decreases by a multiplicative factor $\leq 1/(1+\varepsilon) < 1$. Hence, D' also decreases by a multiplicative factor $\leq 1/(1+\varepsilon)^2 < 1$. Thus, the number of times Step 6 is performed is at most polynomial in $1/\varepsilon$ and the size of the basis.

We showed that the steps of the slide algorithm (Algorithm 7) either preserve or decrease the integer D' by a mulplicative factor < 1 depending on ε. As $D' \geq 1$ and $\log D'$ is initially polynomial in the size of the basis, this means that number of steps for which there is a strict decrease is at most polynomial in $1/\varepsilon$ and the size of the basis. On the other hand, it is not difficult to see that the number of consecutive steps for which D' is preserved is also polynomially bounded: for instance, once Steps 6 are all performed, then all the blocks S_i are HKZ-reduced, which implies that during Step 1, Case 1 cannot occur.

We have seen the main argument why the slide algorithm is polynomial: the number of steps is polynomial. Like in LLL, it would remain to check that all the numbers remain polynomially bounded, which is done in [19]. We only have sketched a proof of the following result:

Theorem 14 ([19]). *There exists an algorithm which, given as input a basis of a d-dimensional integer lattice $L \subseteq \mathbb{Z}^n$, a reduction factor $\varepsilon > 0$, a blocksize $k \geq 2$ dividing d, and access to an SVP-oracle in dimension $\leq k$, outputs a block-Mordell-reduced basis of L with factor ε and blocksize k, such that*

1. *The number of calls to the SVP-oracle is polynomial in the size of the input basis and $1/\varepsilon$.*
2. *The size of the coefficients given as input to the SVP-oracle is polynomial in the size of the input basis.*
3. *Apart from the calls to the SVP-oracle, the algorithm only performs arithmetic operations on rational numbers of size polynomial in the size of the input basis, and the number of arithmetic operations is polynomial in $1/\varepsilon$ and the size of the basis.*

Acknowledgements We thank Nicolas Gama and Damien Stehlé for helpful comments.

References

1. H. Minkowski. *Geometrie der Zahlen*. Teubner, Leipzig, 1896
2. C. L. Siegel. *Lectures on the Geometry of Numbers*. Springer, 1989
3. M. Gruber and C. G. Lekkerkerker. *Geometry of Numbers*. North-Holland, 1987
4. J. Cassels. *An Introduction to the Geometry of Numbers*. Springer, 1997
5. L. Lagrange. Recherches d'arithmétique. *Nouv. Mém. Acad.*, 1773
6. C. Gauss. *Disquisitiones Arithmeticæ*. Leipzig, 1801
7. C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre. *J. Reine Angew. Math.*, 40:279–290, 1850. Also available in the first volume of Hermite's complete works, published by Gauthier-Villars
8. J. Conway and N. Sloane. *Sphere Packings, Lattices and Groups*. Springer, 1998. Third edition
9. S. Khot. *Inapproximability results for computational problems on lattices*. Springer, 2009. In this book
10. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proc. of 15th STOC*, pages 193–206. ACM, 1983
11. R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987
12. G. Hanrot and D. Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In *Advances in Cryptology – Proc. CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 170–186. Springer, 2007
13. G. Hanrot and D. Stehlé. Worst-case Hermite-Korkine-Zolotarev reduced lattice bases. *CoRR*, abs/0801.3331, 2008
14. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd STOC*, pages 601–610. ACM, 2001
15. P. Q. Nguyen and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *J. of Mathematical Cryptology*, 2(2):181–207, 2008
16. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982
17. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987
18. N. Gama, N. Howgrave-Graham, H. Koy, and P. Q. Nguyen. Rankin's constant and blockwise lattice reduction. In *Proc. of Crypto '06*, volume 4117 of *LNCS*, pages 112–130. Springer, 2006
19. N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC '08 – Proc. 40th ACM Symposium on the Theory of Computing*. ACM, 2008.
20. J. Martinet. *Perfect lattices in Euclidean spaces*, volume 327 of *Grundlehren der Mathematischen Wissenschaften*. Springer, Berlin, 2003
21. H. Cohn and A. Kumar. The densest lattice in twenty-four dimensions. *Electron. Res. Announc. Amer. Math. Soc.*, 10:58–67 (electronic), 2004
22. J. Milnor and D. Husemoller. *Symmetric Bilinear Forms*. Springer, 1973
23. D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*. Kluwer Academic Publishers, 2002
24. R. A. Rankin. On positive definite quadratic forms. *J. London Math. Soc.*, 28:309–314, 1953
25. J. L. Thunder. Higher-dimensional analogs of Hermite's constant. *Michigan Math. J.*, 45(2):301–314, 1998
26. M. I. Boguslavsky. Radon transforms and packings. *Discrete Appl. Math.*, 111(1–2):3–22, 2001
27. A. Korkine and G. Zolotareff. Sur les formes quadratiques positives ternaires. *Math. Ann.*, 5:581–583, 1872
28. A. Korkine and G. Zolotareff. Sur les formes quadratiques. *Math. Ann.*, 6:336–389, 1873
29. K. Mahler. A theorem on inhomogeneous diophantine inequalities. In *Nederl. Akad. Wetensch., Proc.*, volume 41, pages 634–637, 1938

30. J. C. Lagarias, H. W. Lenstra, Jr., and C. P. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10:333–348, 1990
31. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *Advances in Cryptology – Proc. EUROCRYPT '08*, Lecture Notes in Computer Science. Springer, 2008
32. L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50. SIAM Publications, 1986. CBMS-NSF Regional Conference Series in Applied Mathematics
33. O. Regev. *On the Complexity of Lattice Problems with Polynomial Approximation Factors*. Springer, 2009. In this book
34. P. Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical report, Mathematische Instituut, University of Amsterdam, 1981. Report 81-04. Available at `http://turing.wins.uva.nl/~peter/`
35. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proc. of 30th STOC*. ACM, 1998. Available at [36] as TR97-047
36. ECCC. `http://www.eccc.uni-trier.de/eccc/`. The Electronic Colloquium on Computational Complexity
37. D. Aharonov and O. Regev. Lattice problems in NP $\cap$ coNP. *J. ACM*, 52(5):749–765 (electronic), 2005
38. O. Goldreich and S. Goldwasser. On the limits of non-approximability of lattice problems. In *Proc. of 30th STOC*. ACM, 1998. Available at [36] as TR97-031
39. I. A. Semaev. A 3-dimensional lattice reduction algorithm. In *Proc. of CALC '01*, volume 2146 of *LNCS*. Springer, 2001
40. P. Q. Nguyen and D. Stehlé. Low-dimensional lattice basis reduction revisited (extended abstract). In *Proc. of the 6th Algorithmic Number Theory Symposium (ANTS VI)*, volume 3076 of *LNCS*, pages 338–357. Springer, 2004. Full version to appear in ACM Transactions on Algorithms, 2009
41. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, 66:181–199, 1994
42. H. W. Lenstra, Jr. Integer programming with a fixed number of variables. Technical report, Mathematisch Instituut, Universiteit van Amsterdam, April 1981. Report 81-03
43. H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1995. Second edition
44. C. Dwork. *Lattices and Their Application to Cryptography*. Stanford University, 1998. Lecture Notes, Spring Quarter. Several chapters are translations of Claus Schnorr's 1994 lecture notes *Gittertheorie und algorithmische Geometrie, Reduktion von Gitterbasen und Polynomidealen*
45. M. Pohst. On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *ACM SIGSAM Bull.*, 15(1):37–44, 1981
46. U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Math. Comp.*, 44(170):463–471, 1985
47. P. Q. Nguyen and D. Stehlé. LLL on the average. In *Proc. of ANTS-VII*, volume 4076 of *LNCS*. Springer, 2006
48. L. J. Mordell. Observation on the minimum of a positive quadratic form in eight variables. *J. London Math. Soc.*, 19:3–6, 1944
49. M. Ajtai. The worst-case behavior of Schnorr's algorithm approximating the shortest nonzero vector in a lattice. In *Proc. 35th Annual ACM Symposium on Theory of Computing*, pages 396–406 (electronic), ACM, 2003