UNIVERSITE DE BOURGOGNE

U.F.R Sciences et Techniques
Institut de Mathématiques de Bourgogne

# Algorithmic approaches to Siegel's fundamental domain

# THESE

présentée et soutenue publiquement le 28 juin 2017

pour l'obtention du grade de

**Docteur de l'Université de Bourgogne**

**(Specialité Mathématiques)**

par

## Carine Jaber

Devant le jury composé des Professeurs

| | | |
|---|---|---|
| COUVEIGNES Jean-Marc | *Université de Bordeaux* | (Président du jury) |
| BRADEN Harry | *Université d'Edimbourg* | (Rapporteur) |
| KOROTKIN Dmitri | *Université Concordia Montréal* | (Rapporteur) |
| ABENDA Simonetta | *Université de Bologne* | (Examinatrice) |
| SCHAUENBURG Peter | *Université de Bourgogne-Franche-Comté* | (Examinateur) |
| KLEIN Christian | *Université de Bourgogne-Franche-Comté* | (Directeur de thèse) |

**Remerciements**

Les remerciements constituent une partie facile et en même temps difficile à écrire. Facile car elle est exclue des corrections mais en même temps un exercice difficile pour les matheux qui n'aiment pas beaucoup parler ni exprimé leur propres sentiments (même si les femmes ne sont pas vraiment reconnues pour cela!).
J'ai toujours considéré la recherche non pas comme un travail mais plutôt comme une passion. Le défi et la curiosité sont les deux mots qui résument mon histoire avec les maths. Ces trois années ont été pour moi une expérience très enrichissante.
J'aimerai commencer par remercier celui qui m'a lancé dans cette fabuleuse aventure, celui avec lequel j'ai pris un très grand plaisir à travailler: mon directeur de thèse Christian Klein que je ne saurai jamais assez remercier pour tout ce qu'il m'a apporté. Je le remercie énormément pour ses précieuses remarques, la qualité de nos discussions mathématiques mais aussi pour sa gentillesse et son écoute. Je tiens également à remercier Peter Schauenburg pour son aide, sa gentillesse et aussi pour sa disponibilité malgré un planning chargé. J'ai eu également la chance et le plaisir de rencontrer Jörg Frauendiener, que je remercie pour son attention, son suivi tout au long de ces trois ans, à distance mais également en n'hesitant pas à venir depuis la Nouvelle Zélande.
Je remercie également Damien Stehlé pour les conseils prodigués et les discussions que nous avons pu avoir. Il me faut également remercier Harry Braden et Dmitri Korotkin pour avoir accepter d'être mes rapporteurs. Merci encore à Simonetta Abenda d'avoir accepter de faire partie du jury et à Jean-Marc Couveignes d'être le président de ce jury.
Faire une thèse, c'est également faire des rencontres!
Je salue et remercie tous les thésards que j'ai pu rencontrer et je leur souhaite une bonne chance pour leur thèses. Je remercie également les membres du laboratoire.
Ce travail n'aurait pu s'effectuer sans la bourse de la société libanaise L'orient SAL que je ne remercierai jamais assez de m'avoir offrir l'opportunité poursuivre mes études en réalisant cette thèse.
Enfin un grand merci à ma famille: ma sœur Léa, mes deux frères et tout particulièrement mes parents qui ont toujours été à mes côtés et fières de leur fille. Je remercie énormément mon amour (mon mari Jamal) et ma princesse (Ella) dont j'étais enceinte durant ma première année de thèse. Ils m'ont

apporté tant de bonheur me permettant d'oublier tout mon stress. Merci à vous tous!!

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

$0^{1\times m-1}$ the column vector of $m-1$ zeros

$0_{n,n}$ the $n \times n$ zero matrix

$\det(A)$ the determinant of a matrix $A$

$\lceil x \rceil$ the smallest integer greater than $x$

$\lfloor x \rceil$ the nearest integer to $x$

$\lfloor x \rfloor$ the greatest integer smaller than $x$

$\mathcal{I}(x)$ the imaginary part of a variable $x$

$\mathcal{R}(x)$ the real part of a variable $x$

$\overline{A}$ the complex conjugate of $A$

$A(:, i)$ the $i$th column of $A$

$A(:, i:j)$ a sub-matrix of A which contains the columns of $A$ from the $i$th to the $j$th position

$A^{-1}$ the inverse of a matrix $A$

$A^{\perp}$ the orthogonal of $A$

$A^{\top}$ the transpose of a matrix $A$

$A_{ij}$ the $i$th row and $j$th column of $A$

$CVP$ the closest vector problem

$GL(n, F)$ the group of all invertible $n \times n$ matrices with entries in $F$

$GSO$ Gram-Schmidt orthogonalization

$HKZ$ Hermite-Korkine-Zolotarev

$i$ the complex unit, $i^2 = -1$

$I_n$ or simply $I$ the $n \times n$ identity matrix

$LLL$ Lenstra, Lenstra and Lovász

$M(n, F)$ the space of all $n \times n$ matrices with entries in the field $F$

$P_n$ the space of positive definite symmetric, real, $n \times n$ matrix

$SVP$ the shortest vector problem

**Abstract**

The action of the symplectic group $Sp(2g, \mathbb{R})$ on Siegel upper half space is a generalization of the action of the group $SL(2, \mathbb{R}) = Sp(2, \mathbb{R})$ on the usual complex upper half plane to higher dimensions. A study of the latter action was done by Siegel in 1943 and led to the well known elliptic fundamental domain. This action for dimension 2, goes back to Gottschling's work in 1959 where 19 conditions were identified to construct this fundamental domain for $g = 2$. However, for $g > 2$, no results appear to be known until now.
The construction of a fundamental domain for the symplectic group is essentially based on the Minkowski reduction theory of positive definite quadratic forms, i.e., a collection of shortest lattice vectors which can be extended to a basis for the lattice and on the maximal height condition ($\det(C\Omega + D) \geq 1$), i.e., find the symplectic element in order to maximize the length of the shortest vector of the lattice generated by the imaginary part of the Riemann matrix. In this work, we present our Minkowski reduction algorithm for dimension $g \leq 5$. Also, we present a part of the finitely many conditions that determine Siegel's fundamental domain for genus 3, especially rank $C = 1$ (the necessity of the conditions is not yet shown).
A motivation for this study is to obtain a rapid convergence of theta functions. The approach adopted here is based on a previous algorithm by Deconinck et al using the Lenstra, Lenstra and Lovász algorithm for finding the shortest lattice vector. In this work, the LLL algorithm is replaced by our exact Minkowski reduction algorithm for small dimensions ($g \leq 5$) and an exact identification of the shortest vector problem for larger values of the genus for two simple reasons: first, the LLL algorithm only provides vectors that are no more than exponentially longer than the shortest ones whereas the Minkowski reduction algorithm finds an exact determination of the shortest lattice vector. Secondly, in the LLL algorithm, there is no reason for the shortest lattice vector to appear at the first position of a matrix whereas it is the first with the Minkowski reduction (an important condition in Siegel's fundamental domain). The utility and effectiveness of this replacement are justified by examples.

**Résumé**

L'action du groupe symplectique $Sp(2n, \mathbb{R})$ sur le demi-espace de Siegel n'est qu'une généralisation de l'action du groupe $SL(2, \mathbb{R}) = Sp(2, \mathbb{R})$ sur le demi-plan de Siegel, à une dimension supérieure. Une étude de cette dernière action a été effectuée par Siegel en 1943 et a conduit au domaine fondamental elliptique bien connu. Cette action pour la dimension 2, remonte au travail de Gottschling en 1959 où 19 conditions ont été identifiés pour construire ce domaine fondamental pour $g = 2$. Cependant, pour $g > 2$, aucun résultat ne semble être connu jusqu'à présent.
La construction d'un domaine fondamental pour le groupe symplectique repose essentiellement sur la théorie de la réduction de Minkowski des formes quadratiques définitives positives, c'est-à-dire une collection de vecteurs les plus courts d'un réseau qui peut être étendue à une base à ce réseau et sur la condition de hauteur maximale ($\det(C\Omega + D) \geq 1$), c'est-à-dire, trouver l'élément symplectique afin de maximiser la longueur du vecteur le plus court d'un réseau engendrée par la partie imaginaire de la matrice de Riemann.
Dans ce travail, nous présentons notre algorithme de réduction de Minkowski pour une dimension $g \leq 5$. De plus, nous présentons une partie des conditions finies qui déterminent le domaine fondamental pour le genre 3, notamment pour rang $C = 1$ (la nécessité des conditions n'est pas encore connue).
Une motivation de cette étude est d'obtenir une convergence rapide des fonctions thêta. L'approche adoptée ici est basée sur un algorithme précédent de Deconinck et al utilisant l'algorithme Lenstra, Lenstra et Lovász pour trouver le vecteur le plus court d'un réseau. Dans ce travail, l'algorithme LLL est remplacé par notre algorithme de réduction de Minkowski pour une petite dimension ($g \leq 5$) et une identification exacte du problème de vecteur le plus court pour des dimensions supérieures pour deux simple raisons: D'abord, l'algorithme LLL ne fournit que des vecteurs qui ne sont pas plus qu' exponentiellement plus longs que les vecteurs les plus courts d'un réseau tandis que l'algorithme de réduction de Minkowski trouve une détermination exacte du vecteur le plus court. Deuxièmement, dans l'algorithme LLL, il n'y a aucune raison pour que le vecteur le plus court apparaisse à la première position d'une matrice alors qu'il est le premier avec la réduction de Minkowski (une condition importante dans le domaine fondamental de Siegel). L'utilité et l'efficacité de ce remplacement sont justifiés par des exemples.

# Chapter 1

# Introduction

The Symplectic group $Sp(2g, \mathbb{R})$ is a generalization of the group $SL(2, \mathbb{R}) = Sp(2, \mathbb{R})$ to higher dimensions. This group acts on a symmetric homogeneous space, called *Siegel upper half space*. This action has a few similarities with the action of $SL(2, \mathbb{R})$ on the *hyperbolic plane* for genus $g = 1$. A study of this action was firstly done by *Carl Ludwig Siegel* in 1943 and published in his book *Symplectic Geometry* [170]. Siegel described a fundamental domain for that action, using Minkowski's reduction of positive definite quadratic forms. It is clear that Siegel's fundamental domain is intimately linked to lattices where the central computational problem on lattices is the shortest vector problem for dimensions greater than 2. The shortest vector problem appears in the context of Siegel's fundamental domain and for this the main part of our work was dedicated to an exact determination of the shortest lattice vector for low-dimensional lattices, more precisely constructing a Minkowski reduced basis for dimensions $\leq 5$. As an application, Siegel's fundamental domain can be used as an efficient method for the computation of theta series.

## 1.1 Lattices and lattice reductions

Lattice reduction plays an important role in many areas of mathematics and computer science (see [28], [139], [91], [65], [108], [85], [183], [83], [128], [138],[107], [105], [43], [143]). Lattice reduction was used to break schemes based on the Knapsack problem[1]. The success of reduction algorithms at breaking various cryptographic schemes over the past twenty years (see [83]) have arguably established lattice reduction techniques as the most popular tool

[1]or the subset sum can be stated as follows: Given a set of $r$ weights $W = (w_0, ..., w_{r-1})$ and a sum $X$, find $x_0$,.., $x_{r-1}$ where each $x_i \in \{0, 1\}$, so that $X = x_0w_0 + x_1w_1 + ... + x_{r-1}w_{r-1}$, see [141]

in public-key cryptanalysis. Ajtai and Dwork in [8] designed a probabilistic public key cryptosystem whose security relies on the hardness of lattice problems. Inspired by the Ajtai-Dwork cryptosystem, Goldreich, Goldwasser and Halevi proposed in [32] a public key cryptosystem based on the closest vector problem in a lattice i.e, the location of the lattice vector closest to a given point $x \in \mathbb{R}^n$ [72] (see also [27]). In [36], Coppersmith showed by means of lattice reduction how to solve rigorously certain apparently non-linear problems related to the question of finding small roots of low-degree polynomials equations. Lattice reduction leads to efficient solutions for several classical problems in lattice theory. For example, lattice reduction is intimately linked to the search for the shortest vector in a lattice, which is a fundamental algorithmic problem and lies at the heart of the solution of many diophantine problems in arithmetics, including integer programming, [101], finding irreducible factors of polynomials (see [88], [45]), minimal polynomials of algebraic numbers, [88], and many more, [172], [5] and [49].
We will show in this first part that an exact determination of the shortest lattice vector can be obtained by lattice reduction algorithms, in particular our Minkowski reduction algorithm. Other algorithms as the LLL algorithm are faster (in polynomial time) then the Minkowski reduction but find the shortest lattice vector merely with an error growing exponentially with the lattice dimension.

**Definition.** A lattice in $\mathbb{R}^g$ is the set

$$\mathcal{L}(b_1, .., b_g) = \left\{ \sum_{i=1}^{g} x_i b_i \mid x_i \in \mathbb{Z} \right\},$$

of all integer combinations of $g$ linearly independent vectors $b_1$,...,$b_g$. These vectors are known as a basis $B$ of the lattice. A lattice basis is usually not unique but all have the same number of elements called "dimension" or "rank" of the lattice. Now, in order to understand when the lattice bases that generate the same lattice hold, we recall that an integer matrix $U$ is called "unimodular" if and only if $|\det(U)| = 1$. Then, it follows

**Lemma.** Let $\mathbf{B}$, $\mathbf{B}' \in \mathbb{R}^{g \times g}$ be two non-singular matrices. One has $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$ if and only if there exists an unimodular matrix $U \in \mathbb{Z}^{g \times g}$ with $\mathbf{B}' = \mathbf{B}U$.

The most fundamental problem involving lattices is to find a "nice" basis, which consists of short and almost orthogonal vectors (see Figure 1.1).

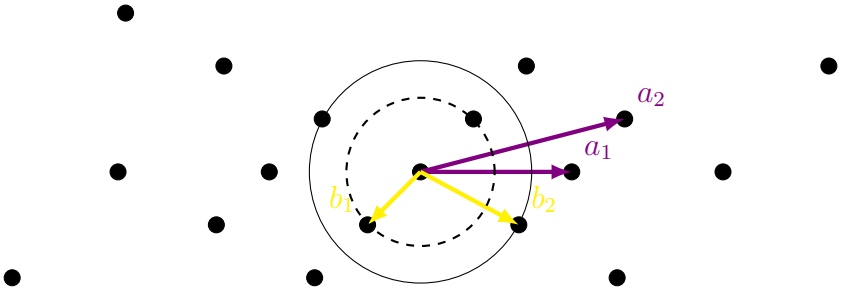


Figure 1.1: The lattice generated by two different bases

The process of improving the quality of a basis by applying well chosen unimodular matrices is generally called **lattice reduction**. The lattice reduction techniques provide a compromise between the quality of the reduced basis and the computational effort required for finding it.

Many different notions of reduced bases exist, and for most of them there is an algorithm for computing a reduced basis from any lattice basis. These reduction algorithms are often classified into two categories according to their complexity: exponential time or the slower algorithms as Minkowski reduced bases and polynomial time where such algorithms are used more from the practical point of view as the LLL algorithm named after Lenstra, Lenstra and Lovász.

In this work, in order to better understand lattice reduction, we study low-dimensional lattices ($g \leq 5$).

### 1.1.1 Gram-Schmidt orthogonalization

The achieved reductions are most often defined in terms of orthogonality. For linearly independent vectors, we define the Gram-Schmidt orthogonalized vectors $b_1^*$,..., $b_g^*$ via an iterative process. First, we define $b_1^* = b_1$ and then for $i = 2, .., g$ we define $b_i^*$ to be the component of $b_i$ orthogonal to $\mathbf{Span}(b_1, .., b_{i-1}) = \mathbf{Span}(b_1^*, .., b_{i-1}^*)$, the linear span of the previous vectors,

$$b_i^* = b_i - \sum_{j<i} \mu_{i,j} b_j^* \text{ where } \mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}.$$

However, the Gram-Schmidt orthogonalization process is not useful in general for lattices since the coefficients $\mu_{i,j}$ do not usually lie in $\mathbb{Z}$ and so the resulting vectors are not elements of the lattice (see, Figure 1.2).
It follows from this definition a simple but not very powerful criterion, the so-called "**size-reduction**"

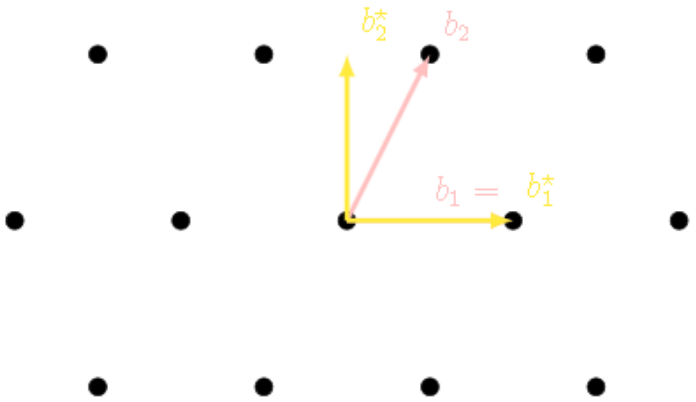


Figure 1.2: GSO

### 1.1.2 Size-reduction

A lattice basis is said to be size-reduced or weakly reduced if its Gram-Schmidt coefficients $\mu_{i,j}$'s all satisfy:

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

The size-reduction was introduced by Lagrange and seems like one of the main condition for several algorithms such as Gauss, HKZ and LLL algorithms.
The size-reduction can be easily achieved by subtracting from each $b_i$ a suitable linear combination $\sum_{j=1}^{i-1} x_j b_j$ of the previous $b_j$'s, for each $i = 2, ..., g$.

### 1.1.3 The Shortest vector problem (SVP)

The most basic computational problem in lattices is the shortest vector problem. The SVP asks to find a non zero lattice vector of smallest norm for a given lattice basis as input. This norm is called **the first minimum**, $\lambda_1(\mathcal{L})$ or the minimum distance and is in general unique up to the sign.
An equivalent way to define the $\lambda_1$: it is the smallest $r$ such that the lattice points inside a ball of radius $r$ span a space of dimension 1.
Minkowski's theorem gives a simple way to bound the length of the shortest lattice vector.

**Theorem** (Minkowski's theorem)**.** $\lambda_1(\mathcal{L}) \leq \sqrt{g}(\det(\mathcal{L}))^{\frac{1}{g}}$ for any $\mathcal{L}$ of dimension $g$.

However, in general $\lambda_1$ can be much smaller than $\sqrt{g}(\det(\mathcal{L}))^{\frac{1}{g}}$ and as an example we take a lattice generated by the following orthogonal vectors

$b_1 = \epsilon e_1$ and $b_2 = (\frac{1}{\epsilon})e_2$ ($e_i$ are the unit vectors). The determinant of this lattice is 1 ($\det(\mathcal{L}) = |\det(B)|$) and we obtain $\lambda_1(\mathcal{L}) \leq \sqrt{2}$ via Minkowski's theorem. However, $\lambda_1(\mathcal{L}) = \epsilon$ can be arbitrarily small.

In fact, the SVP can be solved efficiently by lattice reduction for dimension 2 (Gauss's reduction). However for dimension greater than 2, the exact determination of the shortest lattice vector becomes more complicated to be obtained.

The definition of the first minimum leads to the following generalization of $\lambda_1$ known as "successive minima". The $i$th successive minimum, $\lambda_i(\mathcal{L})$, is the radius of the smallest closed ball centered at the origin containing at least $i$ linearly independent lattice vectors.

**Proposition.** The successive minima of a lattice are always reached, there always exist independent vectors $v_i$'s such that

$$\|v_i\|_2 = \lambda_i(\mathcal{L}), \text{ for all } i.$$

However, for $g > 4$, such vectors do not necessarily form a basis for the lattice. This implies that the construction of a "nice" basis will not be easily obtained.

The best approach for solving the shortest vector problem for low dimensional lattices is the enumeration technique which dates back to the early 1980s with work by Pohst [142], Kannan [87], and Fincke-Pohst [49], and is still actively investigated [159, 5, 70, 144, 158, 130, 174]. These methods are all deterministic and are guaranteed to output a non zero vector of minimum length. The time complexity is exponential in the lattice dimension, but the storage requirements are polynomial. This approach is known by the name "sphere decoding" in the communications community. Enumeration is simply an exhaustive search for the best integer combination of the basis vectors.

Another approach for solving lattice problems was suggested in 2001 by Ajtai, Kumar and Sivakumar [9] (see also [10], [140], [129], [145]). This technique known as sieving algorithm leads to the asymptotically fastest algorithms for solving lattice problems (running in time essentially $2^{O(g)}$) and requires an exponential amount of space. As a result, it is so far not useful in practice.

The basic enumeration algorithm (as in [49, 159]) allows to enumerate all vectors of euclidean norm $\leq R$ (see, Figure 1.3). So, we attempt to search over only lattice points that lie in a certain sphere of radius $R$ around the vector zero, thereby reducing the search space by taking the new radius $R$ equal to the shortest distance between the lattice points and the vector

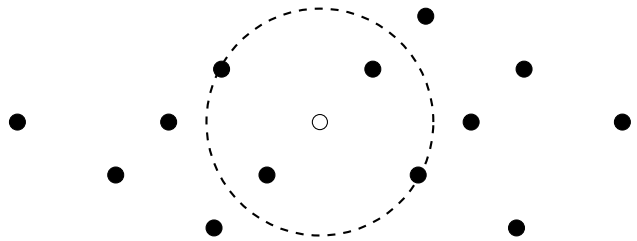

Figure 1.3: Idea behind the sphere decoder for the shortest lattice vector

0. The procedure continues until no points can be found inside the sphere and the last point obtained by this procedure is considered as the shortest lattice vector. The search can be seen as a depth-first on a tree whose leaves correspond to lattice vectors, and whose internal nodes correspond to partial assignments to the coefficients of the integer combination (see, Figure 1.4).

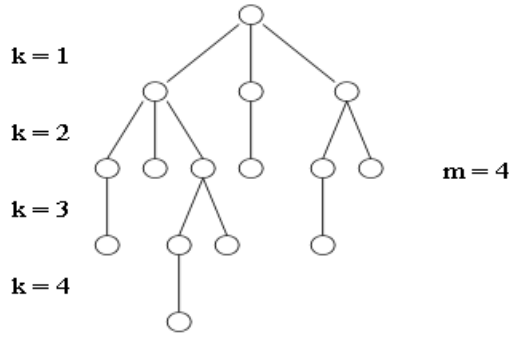


Figure 1.4: Enumeration tree

**The Schnorr-Euchner enumeration.** The Schnorr-Euchner enumeration [159] performs a depth-first search to find the single leaf. The goal is to find the shortest lattice vector, say, $v$. The enumeration goes through the enumeration tree formed by all vectors in the projected lattices $\pi_g(\mathcal{L})$, $\pi_{g-1}(\mathcal{L})$,..., $\pi_1(\mathcal{L})$ of norm at most $R$ where $\pi_i$ is the orthogonal projection on $(b_1, ..., b_{i-1})^{\perp}$. For all $i \in \{1, ..., g+1\}$, $\pi_i(\mathcal{L})$ is a $g+1-i$-dimensional lattice generated by the basis $(\pi_i(b_i), ..., \pi_i(b_g))$. The enumeration tree is a tree of depth $g$, and for each $k \in \{0, ..., g\}$, the nodes at depth $k$ are all the vectors of the rank-$k$ projected lattice $\pi_{g+1-k}(\mathcal{L})$ with norm at most $R$. In particular, the root of the tree is the zero vector (because $\pi_{g+1}(\mathcal{L}) = \{0\}$), while the leaves are all the vectors of $\mathcal{L}$ of norm $\leq R$.
The shortest lattice vector $v \in \mathcal{L}$ can be written as $v = \sum_{i=1}^{g} v_i b_i$ where the $v_i$'s are unknown integers, $\{b_1, ..., b_g\}$ is the lattice basis of $\mathcal{L}$ such that

$b_i = b_i^* + \sum_{j=1}^{i-1} \mu_{ij} b_j^*$. Then $v = \sum_{j=1}^{g} (v_j + \sum_{i=j+1}^{g} \mu_{ij} v_i) b_j^*$ which gives the norm of its projection as:

$$\|\pi_{g+1-k}(v)\|^2 = \sum_{j=g+1-k}^{g} \left( v_j + \sum_{i=j+1}^{g} \mu_{ij} v_i \right)^2 \left\| b_j^* \right\|^2, \quad 1 \leq k \leq g.$$

If $v$ is a leaf of the tree, then we have

$$\sum_{j=g+1-k}^{g} \left( v_j + \sum_{i=j+1}^{g} \mu_{ij} v_i \right)^2 \left\| b_j^* \right\|^2 \leq R^2, \quad 1 \leq k \leq g. \tag{1.1}$$

(1.1) can be written for $1 \leq k \leq g$ as

$$\left| v_{g+1-k} + \sum_{i=g+2-k}^{g} \mu_{ij} v_i \right| \leq \frac{\sqrt{R^2 - \sum_{j=g+2-k}^{g} (v_j + \sum_{i=j+1}^{g} \mu_{ij} v_i)^2 \left\| b_j^* \right\|^2}}{\left\| b_{g+1-k}^* \right\|}. \tag{1.2}$$

Suppose that the projection $\pi_{g+2-k}(v)$ has been computed for some $k$ i.e., the integers $v_{g+1-(k-1)}$,..., $v_g$ are known. Then (1.2) allows to compute an interval $I_{g+1-k}$ such that $v_{g+1-k} \in I_{g+1-k}$, and therefore to perform an exhaustive search for $v_{g+1-k}$. A depth first search of the tree corresponds to enumerating $I_{g+1-k}$ from its middle, by increasing values of $\|\pi_{g+1-k}(v)\|$, namely $v_{g+1-k} = \left\lfloor -\sum_{i=g+2-k}^{g} \mu_{ij} v_i \right\rceil$, $\left\lfloor -\sum_{i=g+2-k}^{n} \mu_{ij} v_i \right\rceil \pm 1$, and so on.

### 1.1.4 Lattice reduction algorithms

The result of the multiplication of the basis $\mathbf{B}$ by a unimodular matrix $U$ can be also obtained by a sequence of elementary column operations on $\mathbf{B}$:

1. Swapping columns: $b_i \leftrightarrow b_j$;
2. Multiplication of a column with $-1$: $-b_i \leftarrow b_i$;
3. Addition of an integral multiple of one column to another column: $b_i \leftarrow b_i + k b_j$ for $i \neq j$ and $k \in \mathbb{Z}$.

Now that we know which operations to apply let us consider an example of lattice basis reduction. Suppose that we want to determine the shortest vector of the lattice generated by the following vectors $u = \begin{bmatrix} 2.1 \\ 1 \end{bmatrix}$ and $v = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$. It is difficult to guess what the shortest vector of this lattice could be. Thus,

we subtract $u$ from $v$ and replace $v$ with the outcome of this operation. The result is this new basis

$$(u, v) = \begin{bmatrix} 2.1 & 0.9 \\ 1 & 0 \end{bmatrix}.$$

Since $\|u\|_2 > \|v\|_2$, we perform a column swap $u \leftrightarrow v$, that leads to the following basis matrix

$$(u, v) = \begin{bmatrix} 0.9 & 2.1 \\ 0 & 1 \end{bmatrix}.$$

Still, it is not easy to determine the shortest vector. Next, we subtract 2-times $u$ from $v$ and replace $v$ with the outcome of this operation, $v = \begin{bmatrix} 0.3 \\ 1 \end{bmatrix}$. Now, according to the figure 1.5, the shortest vector reveals itself easily

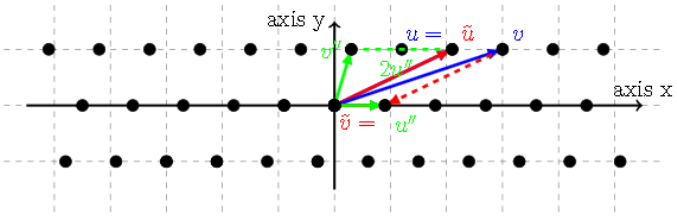


Figure 3.1: Illustration of size reduction (red) and Gauss reduction (green) for a two-dimensional lattice $\mathcal{L}$ spanned by the basis vectors $u = [2.1\ 1]^\top$ and $v = [3\ 1]^\top$ (shown in blue)

Figure 1.5: Gauss's reduction

since the obtained basis consists of two orthogonal vectors. In fact, we have traced here the reduction algorithm of Lagrange which was also described by Gauss. Gauss's algorithm takes a two-dimensional basis matrix $\mathbf{B}$ as input

**Algorithm 1** LagrangeReduce($\mathbf{b}_1$, $\mathbf{b}_2$)

**Input:** A basis $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^2$ for a lattice $L$.
**Output:** A minimal basis $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^2$ of $L$.

1. **repeat**
2. **if** $\|\mathbf{b}_1\| > \|\mathbf{b}_2\|$ **then**
3. swap $\mathbf{b}_1$ and $\mathbf{b}_2$
4. **end if**
5. $\mu_{2,1} := (\mathbf{b}_1 \cdot \mathbf{b}_2) \,/\, \|\mathbf{b}_1\|^2$
6. $\mathbf{b}_2 := \mathbf{b}_2 - \lfloor \mu_{2,1} \rceil \, \mathbf{b}_1$
7. **until** $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|$

Figure 1.6: Gauss's algorithm

and successively performs swaps and size-reduction until a Gauss reduced

basis is obtained $\|b_1\|_2 \leq \|b_2\|_2$. Thus, we have described an algorithm which finds a minimal basis in two dimension and it is similar in style to Euclid's famous gcd (greatest common divisor) algorithm.

For dimension greater than 2, several types of strong reductions are considered in the literature and includes the most frequent one: Minkowski's reduction which gives an exact determination of the shortest lattice vector at least for dimension less or equal to 4 however it is considered as an exponential time algorithm.
In 1905 Minkowski introduced his theory of reduction of positive definite quadratic forms [123]. This theory is one of the essential foundations of the geometry of numbers.

**Definition.** A form $f = \sum_{i,j=1}^{g} a_{ij} x_i x_j$ is Minkowski reduced if for any collection of integers $(l_1, ..., l_g)$, from the condition $\gcd(l_i, ..., l_g) = 1$, it follows that

$$f(l_1, ..., l_g) \geq a_{ii}.$$

A Minkowski reduced basis can also be defined as follows:

**Definition.** A basis $B = \{b_1, .., b_g\}$ of a lattice is Minkowski reduced if for each $i$, $b_i$ is the shortest vector in the lattice such that $b_1$,...,$b_i$ can be extended to a basis of the lattice $\mathcal{L}$.

Note that $(a_{ij})_{1 \leq i,j \leq g} = (\langle b_i, b_j \rangle)_{1 \leq i,j \leq g}$ where we denote by " $\langle ., . \rangle$ " the Euclidean scalar product.

Equivalently;

**Lemma.** A basis $[\mathbf{b}_1, ..., \mathbf{b}_g]_{\leq}$ of a lattice $\mathcal{L}$ is Minkowski-reduced if and only if $i \in [1, .., g]$ and for any integers $x_1, ..., x_g$ such that $x_i, ..., x_g$ are altogether coprime, we have:

$$\|x_1 \mathbf{b}_1 + .... + x_g \mathbf{b}_n\| \geq \|\mathbf{b}_i\| .$$

Fortunately, in any fixed dimension, it is sufficient to check a finite subset of conditions and we call **Minkowski conditions** such a subset with minimal cardinality. Minkowski conditions have been obtained by Tammela up to dimension 7 in [177, 178]. However, it is not clear how to generalize these conditions for higher dimensions. As a consequence, in low dimension, one can check quickly if a basis is Minkowski-reduced by checking these conditions.

Here, we present Tammela's conditions for $g \leq 7$.

**Theorem** (Tammela)**.** Let $g \leq 7$. A basis $[\mathbf{b}_1, ..., \mathbf{b}_g]_{\leq}$ of $\mathcal{L}$ is Minkowski-reduced if and only if for any $i \leq g$ and for any integers $x_1, ..., x_g$ that satisfy both conditions below, we have the inequality:

$$\|x_1\mathbf{b}_1 + .... + x_g\mathbf{b}_g\| \geq \|\mathbf{b}_i\| .$$

1. The integers $x_i, ..., x_g$ are altogether coprime.
2. For some permutation $\nabla$ of $[1, .., g]$, $\left(\left|x_{\nabla(1)}\right|, .., \left|x_{\nabla(g)}\right|\right)$ appears in the list below (where blanks eventually count as zeros).

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 1 | 1 | | | | |
| 3 | 1 | 1 | 1 | | | |
| 4 | 1 | 1 | 1 | 1 | | |
| 5 | 1 | 1 | 1 | 1 | 1 | |
| | 1 | 1 | 1 | 1 | 2 | |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 1 | 1 | 1 | 1 | 2 |
| | 1 | 1 | 1 | 1 | 2 | 2 |
| | 1 | 1 | 1 | 1 | 2 | 3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| | 1 | 1 | 1 | 1 | 1 | 2 | 3 |
| | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| | 1 | 1 | 1 | 1 | 2 | 2 | 3 |
| | 1 | 1 | 1 | 1 | 2 | 2 | 4 |
| | 1 | 1 | 1 | 1 | 2 | 3 | 3 |
| | 1 | 1 | 1 | 1 | 2 | 3 | 4 |
| | 1 | 1 | 1 | 2 | 2 | 2 | 3 |
| | 1 | 1 | 1 | 2 | 2 | 3 | 4 |

Gauss's algorithm computes a Minkowski basis for any two-dimensional lattice. In terms of Minkowski conditions, Gauss's reduction can be defined as follows:

**Definition.** A lattice basis $b_1$, $b_2$ is Gauss reduced if it satisfies

$$\|b_1\| \leq \|b_2\| \leq \|b_1 - b_2\| \leq \|b_1 + b_2\| .$$

For $g > 2$, an algorithm was presented by Helfrich [74] which is polynomial for fixed $g$ but grows exponentially in $g$. Also Phong and Stehlé presented in [137] a new concept of reduced bases called "*greedy-reduced*". They showed that the algorithm takes polynomial time for $g \leq 4$ and this basis found is Minkowski reduced. In fact, there exist many algorithms for producing a Minkowski reduced bases and an HKZ reduced bases in exponential time.

However, up to now, there is no known polynomial time algorithms. Some of these algorithms are more considered as a theoretical result than as a practical tool.

In our work, we present in this context a Minkowski reduction algorithm for $g \leq 5$ inspired by the algorithm presented by Zhang, Qiao and Wei [198].

We search a unimodular matrix $Z$ such that the lattice basis $B' = BZ$ is Minkowski reduced. This work is divided into two parts as indicated in the definition.

1. Constructing the $i$th Minkowski reduced basis vector $m_i$;

2. Extending $\{m_1, ..., m_i\}$ to a basis for the lattice $\mathcal{L}$.

The first part is based on *Schnorr-Euchner enumeration* since enumerative algorithms are more efficient than the probabilistic ones (Sieve algorithms) for $g \leq 40$. However, there is a difference between the enumeration adopted here and the original Schnorr-Euchner enumeration and this is in the way of updating the search radius. In the original one we update the search radius when a shorter lattice vector is found whereas here we update the search radius when a shorter lattice vector satisfying the gcd constraint is found.

For the second part, we use the following results:

**Lemma.** Let $B = \{b_1, ..., b_g\} \in \mathbb{R}^{n \times g}$ and $\mathcal{L}$ be the lattice generated by $B$. For a vector $v = \sum_{i=1}^{g} v_i b_i$ and any index $p$, $1 \leq p \leq g$, there exists a basis for $\mathcal{L}$ containing $\{b_1, .., b_{p-1}, v\}$ if and only if $\gcd(v_p, .., v_g) = 1$.

**Procedure.** Let $[p, q]^\top$ be a non zero integer vector and $\gcd(p, q) = d$. Using the **extended Euclidean** algorithm, we find integers $a$ and $b$ such that $ap + bq = d$. The integer matrix

$$M = \begin{bmatrix} p/d & -b \\ q/d & a \end{bmatrix}$$

is unimodular and

$$M^{-1} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix},$$

such that

$$M^{-1} = \begin{bmatrix} a & b \\ -q/d & p/d \end{bmatrix}.$$

In terms of matrices, we try to find a unimodular matrix $Z$ such that

$$B_{i+1} = B_i Z,$$

where the first $i-1$ columns of $Z$ are the first $i-1$ unit vectors and the $i$th one is the integer vector $z$ obtained by the Schnorr-Euchner enumeration. Therefore, the first $i-1$ columns of $B_{i+1}$ equal to the first $i-1$ columns $m_1$,..,$m_{i-1}$ and the $i$th column of $B_{i+1}$ is $m_i = B_i z$. Since $\gcd(z_i, .., z_g) = 1$ where $z_i, ..., z_g$ are the values taken by $z$ from the $i$th to the last position, we can construct from the above procedure a unimodular matrix $M_i$ whose first column is $[z_i, ..., z_g]^\top$. Therefore, $Z = Z_1 Z_2$ where

$$\mathbf{Z}_1 = \begin{bmatrix} \mathbf{I}_{i-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_i \end{bmatrix}, \quad \mathbf{Z}_2 = \begin{bmatrix} & z_1 & & \\ \mathbf{I}_{i-1} & \vdots & & \mathbf{0} \\ & z_{i-1} & & \\ & 1 & & \\ \mathbf{0} & & \ddots & \\ & & & 1 \end{bmatrix}$$

This algorithm looks like the algorithm presented by Zhang, Qiao and Wei [198] only in the idea and not in the content.

The algorithm is applied to the upper triangular matrix $R$ obtained by the Cholesky decomposition i.e, $Y = R^\top R$. If the initial matrix $Y$ is symmetric positive definite therefore $R = \text{chol}(Y)$ otherwise $R = \text{chol}(Y^\top Y)$.

In fact, Minkowski reduction defines three different domains:

1. The Minkowski domain formed by the set of Minkowski reduced forms considered in the space $R^N$ ($N = \frac{g(g+1)}{2}$) where the domain is a convex cone with finitely many faces;

2. The simple Minkowski domain. This domain can be obtained by adding the condition $a_{i,i+1} \geq 0$ for $i = 1, ..., g-1$;

3. The Minkowski fundamental domain (the "*exact*" domain of reduction) which is contained in a simple domain of Minkowski reduction.

**Definition** ([137])**.** An ordered basis $\{b_1, .., b_g\}_{\leq}$ is Hermite-reduced if it is the smallest basis of $\mathcal{L}$ for the lexicographic order: for any other basis $\left\{b'_1, ..., b'_g\right\}_{\leq}$ of $\mathcal{L}$, we must have $\|b'_1\|_2 = \|b_1\|_2$,..., $\left\|b'_{i-1}\right\|_2 = \|b_{i-1}\|_2$ and $\|b'_i\|_2 > \|b_i\|_2$ for some $1 \leq i \leq g$.

**Definition.** Let $f$ be a Hermite-reduced form, and $S$ be an unimodular matrix. Then by the definition of Hermite reduction, $fS$ is Hermite-reduced if and only if

$$f(S_{i1}, ..., S_{ig}) = a_{ii},\ (i = 1, ..., g).$$

Let $\mathcal{M}$ be the domain of Minkowski reduction (respectively, $\mathcal{H}$ the domain of Hermite reduction); $\mathcal{H} \subset \mathcal{M}$ and the interior points of $\mathcal{H}$ and $\mathcal{M}$ coincide [185]. $\mathcal{H} = \mathcal{M}$ for $g \leq 6$ (see, [153, 177]), $\mathcal{H} \neq \mathcal{M}$ for $g > 6$ [150] (parts of the boundary of $\mathcal{M}$ do not belong to $\mathcal{H}$).
For $g = 2$, the Minkowski fundamental domain is given by the simple Minkowski domain.

**Theorem.**

$$\begin{cases} a_{12} \geq 0 \\ a_{11} - 2a_{12} \geq 0 \\ a_{22} \geq a_{11} > 0 \end{cases} \tag{1.3}$$

All the transformations of a form $f$ ($f$ is a Hermite-reduced form) from the domain (1.3) into reduced form are automorphisms of this form ($M_{f'} = S^{\top} M_f S$). In other words, all the forms of the domain (1.3) are not equivalent.

However, for $g > 2$, the simple Minkowski domain does not define the Minkowski fundamental domain. On the boundary of a simple Hermite-Minkowski domain, there are equivalent points. The fundamental domain for $g = 3$ is given by Tammela (see details in chapter2; section: An introduction to the fundamental domain of Minkowski) but the corresponding conditions in higher dimensions appear to be unknown.

In order to better understand the problem, we consider an example for $g = 3$

**Example.** Consider the face $a_{11} = a_{22}$ of a simple Minkowski domain $\mathcal{M}_0$. This domain intersects with two domains $\mathcal{M}_1$ and $\mathcal{M}_2$ equivalent to it, where $\mathcal{M}_1 = \mathcal{M}_0\tilde{S}_1$, $\mathcal{M}_2 = \mathcal{M}_0\tilde{S}_2$ and $\tilde{S}_1$, $\tilde{S}_2$ are a linear transformations in the space $R^N$ induced by the matrices $S_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $S_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$. Thus, $\mathcal{M}_1 = \begin{pmatrix} a_{22} & a_{12} & a_{23} \\ a_{12} & a_{11} & a_{13} \\ a_{23} & a_{13} & a_{33} \end{pmatrix}$ and $\mathcal{M}_2 = \begin{pmatrix} a_{22} & a_{12} & -a_{23} \\ a_{12} & a_{11} & -a_{13} \\ -a_{23} & -a_{13} & a_{33} \end{pmatrix}$. $\mathcal{M}_0$ and $\mathcal{M}_1$ intersect along the part $a_{13} \geq 0$ and $\mathcal{M}_0$ and $\mathcal{M}_2$ intersect along the part

$a_{13} \leq 0$ of the face $a_{11} = a_{22}$.
It can be easily seen that the partition of the cone of positivity into domains equivalent to a simple Minkowski domain is not normal. Some domains intersect along pieces of the faces. The idea of Tammela here was to reconstruct a new partition of all the faces of a simple Minkowski domain where all the faces are not equivalent.

Now, we present 2 systems among the 16 systems that define the Minkowski fundamental domain for $g = 3$ where only the boundary points are considered in this case.

$$\begin{cases} a_{12} = 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + \\ 2a_{13} - 2a_{23} > 0 \\ a_{13} \geq 0 \end{cases} \qquad \begin{cases} a_{12} = 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} = a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{23} \geq a_{13} \\ a_{13} \geq 0 \end{cases}$$

The famous one in the second category of lattice reductions is the LLL algorithm. Lenstra, Lenstra and Lovász introduced the notion of LLL reduction in their article [103] where an algorithm for computing such reduced bases is also presented. The LLL algorithm can be seen as an extension of Gauss reduction for dimension greater than 2. The LLL algorithm generalizes Gauss reduction and exploits the Gram-Schmidt orthogonalization.

**Definition.** A lattice basis $B = \{b_1, ...., b_g\}$ is LLL reduced if:

- It is size reduced, i.e, $|\mu_{ik}| \leq \frac{1}{2}$ for $1 \leq i < k \leq g$;
- $\delta \left\| b_{i-1}^* \right\|^2 \leq \left\| b_i^* \right\|^2 + \mu_{i,i-1}^2 \left\| b_{i-1}^* \right\|^2$,

such that $\frac{1}{4} < \delta \leq 1$ and $\left\{ b_1^*, ...b_g^* \right\}$ is the Gram-Schmidt orthogonalization of $B$.

The LLL algorithm performs a sequence of steps: translations and swap steps, where the aim of translations is to satisfy the size-reduction condition and the swap is to shift the weight in the orthogonalized basis $B^*$ from the first to the last ones.

However, this type of reduction has some drawbacks:
The first vector in an LLL-reduced basis approximates a shortest lattice vector.

**Corollary 1.** In an LLL-reduced basis $\mathbf{B}$ with $\delta = 3/4$, we have

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2}.\lambda_1(\mathcal{L}(\mathbf{B})).$$

Hence, we can deduce that

- The LLL algorithm runs in **polynomial time** but it only provides vectors that are no more than **exponentially** longer (in $n$) than the shortest ones.
- In the LLL algorithm, there is no reason for the shortest lattice vector to appear at the first position of a matrix. An important remark for Siegel's fundamental domain where the shortest lattice vector is assumed to appear at the fist position of a matrix (see, [51] and chapter 5).

Now, to illustrate the difference between our Minkowski reduction algorithm for dimension less or equal to 5 and the LLL algorithm, we consider examples of random matrices,

```
Y=
  1.4903   0.4132  -0.2758   0.6514
  0.4132   1.1509  -0.2646   0.0255
 -0.2758  -0.2646   0.8102   0.2585
  0.6514   0.0255   0.2585   0.6964.
```

This matrix was created using a $4 \times 4$ matrix $L$ with random entries, and then setting $Y = L^\top L$. We put $Y = R^\top R$ where the upper triangular matrix $R$ is obtained from $Y$ via a Cholesky decomposition (see, Definition 3). To this matrix $R$, we apply our Minkowski reduction algorithm and this leads to the following Minkowski reduction matrix (it suffices to check Minkowski's conditions to prove it)

```
  0.6255   0.2135   0.2759   0.1231
  0.2135   0.6964   0.2585   0.0255
  0.2759   0.2585   0.8102  -0.2646
  0.1231   0.0255  -0.2646   1.1509.
```

The diagonal elements of a Minkowski reduced matrix represent the norm of the shortest vectors in ascending order.
The corresponding LLL reduced matrix ($\delta = 3/4$) takes the form

```
 0.8102  -0.2759   0.2585  -0.2646
-0.2759   0.6255  -0.2135  -0.1231
 0.2585  -0.2135   0.6964   0.0255
-0.2646  -0.1231   0.0255   1.1509.
```

This example shows that the LLL algorithm gives the shortest lattice vector as a second vector in contrast to a Minkowski ordered matrix where the shortest vector is always the first of the matrix.

The second example shows the overestimation of the length of the shortest vector even for small size of the matrix

```
Y=
0.7563   0.4850  0.4806  0.3846
0.4850   1.3631  0.2669 -0.3084
0.4806   0.2669  0.7784 -0.4523
0.3846  -0.3084 -0.4523  1.7538.
```

The Minkowski reduction yields

```
 0.5321    0.2058   -0.1639    0.0181
 0.2058    0.5735    0.0920    0.2634
-0.1639    0.0920    0.5741    0.1364
 0.0181    0.2634    0.1364    0.6535.
```

It can be seen that the squared length of the shortest lattice vector is 0.5321, the (11) element of the matrix. However, an LLL reduction with $\delta = 3/4$ of the matrix $Y$ leads to

```
 0.7563   -0.2757    0.3182   -0.1089
-0.2757    0.5735    0.0920    0.2634
 0.3182    0.0920    0.5741    0.1364
-0.1089    0.2634    0.1364    0.6535.
```

We notice here that the length of the shortest vector identified by the LLL reduction is 0.5735, longer than the shortest vector obtained by the Minkowski reduction (0.5321) and appears as the second vector in contrast to the Minkowski matrix where an exact determination of the length of the shortest vector appears at the first element of the matrix.

## 1.2 Siegel's fundamental domain

The construction of the fundamental domain for the modular symplectic group is essentially based on the Minkowski reduction theory of positive quadratic forms. In the matrix language, the problem of reduction of positive quadratic forms relative to unimodular equivalence is that of construction of a fundamental domain for the unimodular group $\iota = \mathbf{GL}(g, \mathbb{Z})$ acting on the cone

$$\mathbf{P} = \mathbf{P}_g := \left\{ Y \in M(g, \mathbb{R}) \mid Y^\top = Y, Y \text{ positive definite} \right\}$$

of real positive definite matrices of order $g$ by

$$\iota \ni V : Y \mapsto Y[V] := V^\top Y V.$$

In other words, $Y$ must be in the Minkowski fundamental domain: we search the fundamental domain of the unimodular group such that the matrix $V^\top Y V$ is Minkowski-reduced.
We will see that Siegel's fundamental domain shows that every Riemann surface is $\Gamma_g$-equivalent to a unique point of the fundamental region. It also permits to decide whether two different algebraic curves define the same Riemann surface and this by constructing for both Riemann matrices the symplectic transformation to the Siegel fundamental domain. If both Riemann matrices map to the same point in the fundamental domain, they correspond to the same surface.
We begin this part with some definitions.

**Definition 1.** *Siegel's modular group* $\Gamma_g$ is the group of all $2g \times 2g$ matrices with integer entries which satisfy the condition

$$M^\top J M = J,$$

where $M \in \Gamma_g$, and if $I_g$ and $0_g$ are the $g \times g$ unit and zero matrices respectively, and where

$$J = \begin{pmatrix} 0_g & I_g \\ -I_g & 0_g \end{pmatrix}.$$

It follows from the definition that a $2g \times 2g$ integer matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ with $g \times g$-blocks $A$, $B$, $C$, $D$ is symplectic if and only if

$$A^\top C = C^\top A,\ B^\top D = D^\top B,\ A^\top D - C^\top B = I_g. \tag{1.4}$$

It is easy to see that a matrix $M$ is symplectic if and only if the matrix $M^\top = JM^{-1}J^{-1}$ is symplectic. This implies that the conditions (1.4) can be rewritten in the form

$$AB^\top = BA^\top,\ CD^\top = DC^\top,\ AD^\top - BC^\top = I_g.$$

Generators for $\Gamma_g$ were first determined by Hua and Reiner [79]. In 1961, Klingen [90] obtained a characterization of $\Gamma_g$ for $g \geq 2$ by a finite system of defining relations. In 1983, Mumford [133] gave three generators for the modular group $\Gamma_g$.

$$\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix},\ \begin{pmatrix} A & 0 \\ 0 & (A^{-1})^\top \end{pmatrix},\ \begin{pmatrix} I & B \\ 0 & I \end{pmatrix}$$

for all $A \in GL(g, \mathbb{Z})$ and $B$ a symmetric, integer matrix.

**Definition 2.** A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive definite if

$$x^\top Ax > 0 \text{ for all } x \neq 0.$$

Note that if $A$ is an $n \times n$ symmetric matrix, then $x^\top Ax$ is the function

$$x^\top Ax = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}x_ix_j = \sum_{i=1}^{n} A_{ii}x_i^2 + 2\sum_{i>j} A_{ij}x_ix_j;$$

this is called "a quadratic form".

**Definition 3** (Cholesky Factorization)**.** Every positive definite matrix $A \in \mathbb{R}^{n\times n}$ can be factored as

$$A = R^\top R$$

where $R$ is upper triangular matrix with positive diagonal elements.

**Definition 4.** A matrix $\Omega \in M(g, \mathbb{C})$ is called a *Riemann matrix*, if it is symmetric and its imaginary part $\mathcal{I}(\Omega)$ is positive definite.

**Definition 5.** The set of Riemann matrices is denoted by $H^g$ and generally called *the Siegel upper half space* of degree $g$ (or genus $g$).

It was introduced by Siegel in 1939 and obviously

$$H^g \cong \mathbb{R}^{\frac{g(g+1)}{2}} \times P_g \subset \mathbb{R}^{g(g+1)},$$

The group $Sp(2g,\mathbb{R})$ operates on $H^g$ by

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}.\Omega := (A\Omega + B)(C\Omega + D)^{-1},$$

where $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in Sp(2g,\mathbb{R})$,
one has

$$\mathcal{I}(M.\Omega) = (C\Omega + D)^{-\top}\mathcal{I}(\Omega)(C\Omega + D)^{-1}.$$

We give a brief overview of this action. The facts stated here will be given in more detail later.

We consider the Siegel modular group $\Gamma_g = Sp(2g,\mathbb{Z})$ of degree $g$ and the generalized upper half plane $H^g$, on which $\Gamma_g$ acts properly discontinuously. For $\Gamma_g$, a fundamental domain on $H^g$, which (roughly) contains one representative from the orbit of every $\Omega \in H^g$, has been given in [171].

**Definition 6.** Siegel's fundamental domain is the subset of $H^g$ such that $\Omega = X + iY \in H^g$ satisfies:

1. $|X_{jk}| \leq \frac{1}{2}$, $j,k = 1,...,g$;
2. $Y$ is in the fundamental region of Minkowski reductions;
3. $|\det(C\Omega + D)| \geq 1$ for all $C$, $D$.

Note that the third condition must be verified for all matrices $C$ and $D$ of the symplectic group and it is also called the maximal height condition.

For genus 1, Siegel's fundamental domain is given by the well-known "elliptic fundamental domain"

$$D := \left\{ \Omega \in H^1 \mid |\Omega| \geq 1, |\mathcal{R}(\Omega)| \leq \frac{1}{2} \right\}.$$

In fact, there are many ways of constructing a fundamental domain (see, Figure 1.7) whereas a common choice is the above region, bounded by the vertical line $\mathcal{R}(\Omega) = \frac{1}{2}$ and $\mathcal{R}(\Omega) = -\frac{1}{2}$ and the circle $|\Omega| = 1$. It has vertices at $\frac{1}{2} + i\frac{\sqrt{3}}{2}$ and $-\frac{1}{2} + i\frac{\sqrt{3}}{2}$, where the angle between its sides is $\frac{\pi}{3}$ and a third vertex at infinity, where the angle between its sides is 0.
Since the points on the borders of this region are equivalent under symplectic transformations: the points on the two lines $\mathcal{R}(\Omega) = \pm\frac{1}{2}$ are equivalent under the action of $T : \omega \to \Omega \pm 1$ and the points on the left and right halves of the arc $|\Omega| = 1$ are also equivalent under the action $S : \Omega \to -\frac{1}{\Omega}$, it is better to

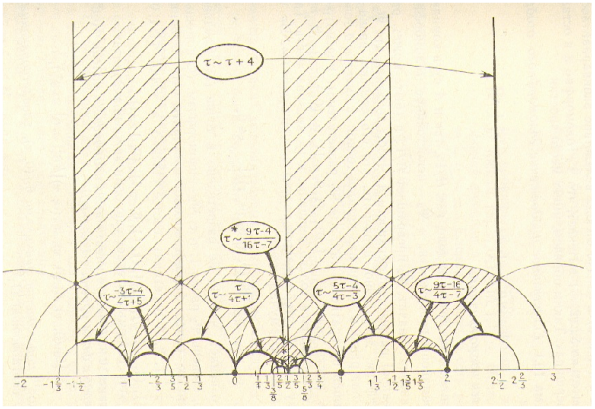

Figure 1.7: The elliptic fundamental domain [Mumford]

consider a part of the boundary of this domain more precisely, we add only the boundary points with non-positive real part. For this reason, we define the semi-closure of D as the fundamental region of the elliptic case. Thus, every point of $H^g$ is equivalent under modular group to a unique point of this new fundamental domain.

We can also factorize $H^2$ by the action of $Sp(4,\mathbb{Z})$ $(g=2)$. This gives the Siegel-Gottschling fundamental domain $H^2/Sp(4,\mathbb{Z})$. In this case, the fundamental domain is defined by the following set of inequalities:
The standard bounds on the real part of the Riemann matrix $\Omega$,

$$|X_{11}| \leq \frac{1}{2},\ |X_{12}| \leq \frac{1}{2},\ |X_{22}| \leq \frac{1}{2},$$

the simple Minkowski ordering conditions on $Y = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{12} & Y_{22} \end{pmatrix}$, the imaginary part of $\Omega$:

$$Y_{22}, Y_{11} \geq 2Y_{12},\ Y_{22} \geq Y_{11},\ Y_{12} \geq 0$$

and the following 19 inequalities corresponding to the third conditions:

$$|\Omega_{11}| \geq 1,\ |\Omega_{22}| \geq 1,\ |\Omega_{11} + \Omega_{22} - 2\Omega_{12} + e| \geq 1,$$

these consitions correspond to rank $C = 1$ where $e = \pm 1$, and for rank $C = 2$ since $C = I_2$, the third condition is replaced by $|\det(\Omega + S)| \geq 1$ where $S$ are the matrices

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{e} & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{e} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{e} & 0 \\ 0 & \mathbf{e} \end{pmatrix},$$
$$\begin{pmatrix} \mathbf{e} & 0 \\ 0 & -\mathbf{e} \end{pmatrix}, \quad \begin{pmatrix} 0 & \mathbf{e} \\ \mathbf{e} & 0 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{e} & \mathbf{e} \\ \mathbf{e} & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & \mathbf{e} \\ \mathbf{e} & \mathbf{e} \end{pmatrix}. \tag{1.5}$$

A constraint appears in the construction of such a domain: the maximal height condition that must be verified for all matrices $C$ and $D$ of the symplectic group. For this reason, Siegel showed in his book [171] that third condition is equivalent to a finite number of conditions i.e., just a finite number of matrices has to be considered. However, it is not trivial to determine these set of matrices for dimension greater than 2.
We present in this context, a part of the finitely many inequalities that determine the fundamental domain for $g = 3$, especially for rank$C = 1$ (see, Chapter 5).

**Theorem 1.** For genus 3 and a rank 1 matrix $C$, we have the following inequalities to be verified:

Writing $\Omega = \begin{pmatrix} \Omega_1 & \Omega_4 & \Omega_5 \\ \Omega_4 & \Omega_2 & \Omega_6 \\ \Omega_5 & \Omega_6 & \Omega_3 \end{pmatrix}$ then

$$|\Omega_1| \geq 1,\ |\Omega_2| \geq 1,\ |\Omega_3| \geq 1,\ |\Omega_1 + \Omega_2 - 2\Omega_4 \pm 1| \geq 1, |\Omega_2 + \Omega_3 - 2\Omega_6 \pm 1| \geq 1,$$

$$|\Omega_1 + \Omega_3 + 2\Omega_5 \pm 1| \geq 1 \text{ if } \mathcal{I}(\Omega_5) < 0,\ |\Omega_1 + \Omega_3 - 2\Omega_5 \pm 1| \geq 1 \text{ if } \mathcal{I}(\Omega_5) > 0,$$

$$|\Omega_1 + \Omega_2 + \Omega_3 + 2\Omega_4 - 2\Omega_5 - 2\Omega_6 + d| \geq 1,\ \text{for } -4 \leq d \leq 4,\ \text{and } \mathcal{I}(\Omega_5) > 0,$$

$$|\Omega_1 + \Omega_2 + \Omega_3 - 2\Omega_4 - 2\Omega_5 + 2\Omega_6 + d| \geq 1, \text{for } -4 \leq d \leq 4,\ \text{and } \mathcal{I}(\Omega_5) > 0,$$

$$|\Omega_1 + \Omega_2 + \Omega_3 - 2\Omega_4 + 2\Omega_5 - 2\Omega_6 + d| \geq 1,\ \text{for } -4 \leq d \leq 4.$$

For this result, we have used Tammela's conditions for constructing a Minkowski fundamental domain for dimension 3 [176] and the following lemma due to Gottschling.

**Lemma** (Gottschling)**.** In the domain $\mathcal{B}_e$ where $e = -1, 0, 1$, the inequality

$$|\mathbb{B}_{11} + \mathbb{B}_{22} - 2\mathbb{B}_{33} - 2e| \geq 1;$$

is a sequence of $|\det(\mathbb{B} + eS)| \geq 1$ with $S = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

$\mathcal{B}_0$ is the part of $\mathcal{B}$ in which $Y_{22} \leq 1$, $\mathcal{B}_1$ is a part of $\mathcal{B}_0$ in which $-\frac{1}{2} \leq X_3 \leq -\frac{1}{4}$ and $\mathcal{B}_{-1}$ that part of $\mathcal{B}_0$ in which $\frac{1}{4} \leq X_3 \leq \frac{1}{2}$. $\mathcal{B}$ is a region that contains the standard limits for the real part and the simple Minkowski conditions for dimension 2 and the two inequalities $|\mathbb{B}_{11}| \geq 1$ and $|\mathbb{B}_{22}| \geq 1$.

Now, since the Minkowski fundamental domain appearing in the second condition is only known for $g \leq 3$ (see, [176]) and the third condition is the least studied one, there is no constructive approach to actually identify the domain for $g > 2$. However, Siegel [171] gave an algorithm to approximately reach the fundamental domain called **Siegel reduction**. Siegel's reduction preserves the first condition presented in its fundamental domain and among the finitely many inequalities of the third condition, Siegel's reduction is concerned with that the absolute value of the first element of the Riemann matrix must be greater or equal to 1, i.e, $|\Omega_{11}| \geq 1$.

**Theorem 2** (Siegel Reduction)**.** Every Riemann matrix $\Omega$ can by means of a symplectic transformation be reduced to a Riemann matrix $\tilde{\Omega} = \tilde{X} + i\tilde{Y}$, where $\tilde{Y} = R^\top R$, $\tilde{X}(\tilde{Y})$ are the real (imaginary) part of $\tilde{\Omega}$ and $R$ is an upper triangular matrix, obtained by *Cholesky decomposition*, such that

1. $\left|\tilde{X_{jk}}\right| \leq \frac{1}{2}$, for $j, k = 1, ..., g$;
2. The length of the shortest lattice vector of the lattice generated by the columns of $R$ is bound from below by $\sqrt{\sqrt{3}/2}$.

The second condition is ensured by the following quasi-inversion:

$$\Omega \mapsto \tilde{\Omega} := (A\Omega + B)(C\Omega + D)^{-1} \tag{1.6}$$

such that

$$A = \begin{pmatrix} 0 & 0_{n-1}^\top \\ 0_{n-1} & I_{n-1} \end{pmatrix}, \; B = \begin{pmatrix} -1 & 0_{n-1}^\top \\ 0_{n-1} & 0_{n-1,n-1} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0_{n-1}^\top \\ 0_{n-1} & 0_{n-1,n-1} \end{pmatrix}, \; D = \begin{pmatrix} 0 & 0_{n-1}^\top \\ 0_{n-1} & I_{n-1} \end{pmatrix}$$

Where does the second condition comes from?
Siegel showed that the determinants of the imaginary parts of two Riemann matrices connected by a modular transformation (1.6) are related by

$$\left|\det(\tilde{Y})\right| = \frac{|\det(Y)|}{|\det(C\Omega + D)|^2}, \tag{1.7}$$

then by using (1.6) with the above $A$, $B$, $C$ and $D$, (1.7) becomes

$$\left|\det(\tilde{Y})\right| = \frac{|\det(Y)|}{|\Omega_{11}|^2}$$

and finally by the maximal height condition we obtain the second condition of Siegel's reduction.

A motivation for this work will stem from **theta functions** associated to Siegel upper half space, more precisely an efficient method for the computation of these theta functions by using Siegel's reduction will be presented in this part.

## 1.3 Theta Functions

The theta function of a Riemann surface is a very fundamental special function associated to a Riemann surface both in algebro-geometrical calculations and in applications to non-linear equations and cryptography [175, 82, 53, 54]. Partial solutions of differential equations can often be written in terms of abelian functions, and this in terms of theta functions [93]. Jacobi introduced theta functions in his study of elliptic functions. The Riemann theta function was introduced by Riemann as a generalization of Jacobi's theta functions of one variable for solving the Jacobi inversion problem on general compact connected Riemann surfaces [149, 1].
Multidimensional theta functions are functions in $g$ complex variables and a convenient tool to work with meromorphic functions. In general, each Riemann surface is characterized by its Riemann matrix [39] and to a Riemann matrix one can associate a Riemann theta function.
Note that in general, the presence of symmetries allows to significantly simplify the Riemann matrix of a surface, but only in a homology basis adapted to the symmetries, see [23] for the Klein curve.
We start by some basic facts about theta functions and we define them as an infinite series.

**Definition.** Let $\Omega$ be a $g \times g$ Riemann matrix. The theta function with characteristic $[p, q]$ is defined as

$$\theta_{pq}(z, \Omega) = \sum_{N \in \mathbb{Z}^g} \exp\left\{ i\pi \left\langle \Omega(N+p), N+p \right\rangle + 2\pi i \left\langle z+q, N+p \right\rangle \right\}, \quad (1.8)$$

with $z \in \mathbb{C}^g$ and $p, q \in \mathbb{C}^g$, where $\langle ., . \rangle$ denotes the Euclidean scalar product $\langle N, z \rangle = \sum_{i=1}^{g} N_i z_i$.

The properties of the Riemann matrix ensure that the series converges absolutely and that the theta function is an entire function on $\mathbb{C}^g$. The theta function with characteristic is related to the Riemann theta function $\theta$, the

theta function with zero characteristic $\theta := \theta_{00}$, via

$$\theta_{pq}(z,\Omega) = \theta(z+\Omega p+q)\exp\left\{i\pi\left\langle \Omega p,p\right\rangle + 2\pi\left\langle p,z+q\right\rangle\right\}. \qquad (1.9)$$

The theta function has the periodicity properties

$$\theta_{pq}(z+e_j) = e^{2\pi p_j}\theta_{pq}(z),\ \theta_{pq}(z+\Omega e_j) = e^{-2\pi(z_j+q_j)-i\pi\Omega_{jj}}\theta_{pq}(z), \qquad (1.10)$$

where $e_j$ is a vector in $\mathbb{R}^g$ consisting of zeros except for a 1 in the $j$th position. These periodicity properties (1.10) can be used in the computation of the theta function: an arbitrary vector $z \in \mathbb{C}^g$ can be written in the form $z = \hat{z} + N + \Omega M$ with $N,\ M \in \mathbb{Z}^g$, where $\tilde{z} = \Omega\hat{p} + \hat{q}$ with $|\hat{p}_i| \leq \frac{1}{2}$, $|\hat{q}_i| \leq \frac{1}{2}$. Thus, it is sufficient to compute the theta function for arguments $\tilde{z}$ lying in the fundamental domain of the Jacobian. For general arguments $z$ one computes $\theta(\hat{z},\Omega)$ and obtains $\theta(z,\Omega)$ from the periodicity properties (1.10) by multiplying with an appropriate exponential factor.
In this work, we are interested in a rapid convergence of these multi-dimensional theta functions. The basic idea is to approximate the expression (1.8) via a truncated series.
To compute the series (1.9), it will be approximated by a sum, $|N_i| \leq N_\epsilon$, $i = 1,...,g$ where the constant $N_\epsilon$ is chosen such that all omitted terms in (1.8) are smaller than some prescribed value of $\epsilon$. In contrast to [38], we do not give a specific bound for each $N_i$, $i = 1,...,g$, i.e., we sum over a $g$-dimensional sphere instead of an ellipsoid. Taking into account that we can choose $z$ in the fundamental domain of the Jacobian, we get for the Riemann theta function the estimate

$$N_\epsilon > \sqrt{-\frac{\ln\epsilon}{\pi y_{min}}} + \frac{1}{2}$$

such that

$$y_{min} = \left\langle N_{min}, N_{min}\right\rangle_Y := \min_{N\in\mathbb{Z}^g/0}\left\langle YN,N\right\rangle,$$

i.e., $y_{min}$ is the shortest vector of the lattice generated by the imaginary part $Y$ of the Riemann matrix $\Omega$.
Thus, the longer the shortest vector, the more rapid the convergence of the theta series. Changing the shortest vector can be achieved by changing the homology basis of the underlying Riemann surface which yields a different but symplectically equivalent Riemann matrix. An important step in the efficient computation of theta functions is the construction of appropriate symplectic transformations to generate larger norms of the shortest vector for a given Riemann matrix $\Omega$ assuring a rapid convergence of the theta series.

The behavior of theta functions under modular transformations is explicitly known. One has

$$\theta_{\tilde{p}\tilde{q}}(\mathcal{M}^{-1}z,\tilde{\mathbb{B}}) = k\sqrt{\det(\mathcal{M})}\exp^{\frac{1}{2}\sum_{i\leq j} z_i z_j \frac{\partial}{\partial \mathbb{B}_{ij}} \ln\det \mathcal{M}}\theta_{pq}(z),$$

$$\mathcal{M} = C\mathbb{B} + D,\ \begin{pmatrix}\tilde{p}\\ \tilde{q}\end{pmatrix} = \begin{pmatrix} D & -C\\ -B & A\end{pmatrix}\begin{pmatrix}p\\ q\end{pmatrix} + \frac{1}{2}\begin{pmatrix}\mathrm{diag}(CD^\top)\\ \mathrm{diag}(AB^\top)\end{pmatrix}.$$

Thus, such transformations can dramatically increase the rate of convergence which is especially important for larger values of $g$.

Now, the goal is to find these symplectic transformations on the Riemann matrix $\Omega$ and this will be done by Siegel's fundamental domain. We have already seen that Siegel has constructed a fundamental region for Riemann matrices, analogous to the elliptic case and the algorithm finds iteratively a new Riemann matrix. However, the determination of such a domain becomes more complicated for $g > 2$.
This approach was for the first time implemented in an algorithm by Deconinck et al. in [38]. This algorithm is based on Siegel's reduction together with the LLL algorithm for finding the shortest lattice vector. In our work, the LLL algorithm is replaced by our Minkowski reduction algorithm for dimension $\leq 5$ and an exact determination of the shortest lattice vector for higher dimensions.

In order to generate larger norms of the shortest lattice vector, [38] adopted the following algorithm:
1. Apply the LLL algorithm on the imaginary part of the Riemann matrix in order to find the shortest lattice vector **as the first vector** of the lattice.
2. Subtract an integer matrix to $\Omega$ so that $|\mathcal{R}(\Omega_{ij})| \leq \frac{1}{2}$.
3. If $|\Omega_{11}| \geq 1$, terminate the algorithm; if not apply a quasi-inversion $\Omega \mapsto \tilde{\Omega} = (A\Omega + B)(C\Omega + D)^{-1}$ such that

$$A = \begin{pmatrix} 0 & 0_{n-1}^\top\\ 0_{n-1} & I_{n-1}\end{pmatrix},\ B = \begin{pmatrix} -1 & 0_{n-1}^\top\\ 0_{n-1} & 0_{n-1,n-1}\end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0_{n-1}^\top\\ 0_{n-1} & 0_{n-1,n-1}\end{pmatrix},\ D = \begin{pmatrix} 0 & 0_{n-1}^\top\\ 0_{n-1} & I_{n-1}\end{pmatrix},$$

then go back to step 1 for the resulting $\Omega$.

It is clear that the shortest lattice vector is assumed to appear at the first position of the imaginary part of the Riemann matrix.

Now, in order to show the improvement that can be achieved in the computation of theta series if we replace the LLL algorithm used by Deconinck et al by our Minkowski reduction algorithm for small dimensions and an exact determination of the shortest vector problem for higher dimensions, we take

as an example (among many others) the Riemann matrix of the Fricke-Macbeath surface [56, 115], a surface of genus $g = 7$ with the maximal number $84(g-1) = 504$ of automorphisms. It can be defined via the algebraic curve

$$f(x, y) := 1 + 7yx + 21y^2x^2 + 35x^3y^3 + 28x^4y^4 + 2x^7 + 2y^7 = 0.$$

```
RieMat =

  Columns 1 through 4

   1.0409 + 1.3005i   0.0530 + 0.3624i   0.3484 + 0.0000i
   0.0530 + 0.3624i  -0.5636 + 1.0753i   0.0187 - 0.5975i
   0.3484 + 0.0000i   0.0187 - 0.5975i   1.0544 + 1.7911i
   0.2077 + 0.6759i   0.6749 + 0.3001i   0.3220 - 1.0297i
  -0.2091 - 0.2873i   0.1220 - 0.5274i   0.3029 + 0.8379i
  -0.1064 - 0.4257i   0.1205 - 0.1783i  -0.2297 - 0.3668i
   0.3590 + 0.5023i   0.1990 - 0.1118i   0.3495 - 0.0499i

  Columns 5 through 7

   0.2077 + 0.6759i  -0.2091 - 0.2873i  -0.1064 - 0.4257i
   0.6749 + 0.3001i   0.1220 - 0.5274i   0.1205 - 0.1783i
   0.3220 - 1.0297i   0.3029 + 0.8379i  -0.2297 - 0.3668i
  -0.0978 + 1.7041i  -0.7329 - 0.8055i  -0.0714 - 0.1766i
  -0.7329 - 0.8055i   1.1824 + 1.0163i   0.4425 + 0.2592i
  -0.0714 - 0.1766i   0.4425 + 0.2592i   0.2815 + 0.7791i
  -0.0415 + 0.5448i   0.0835 - 0.2430i  -0.6316 - 0.0369i

  Columns 7 through 7

   0.3590 + 0.5023i
   0.1990 - 0.1118i
   0.3495 - 0.0499i
  -0.0415 + 0.5448i
   0.0835 - 0.2430i
  -0.6316 - 0.0369i
   0.2315 + 0.6895i.
```

We notice that after the LLL reduction, the first vector of the lattice presented as the first component of the imaginary part of the above Riemann

matrix has the squared norm 1.3005. Since the norm of the shortest vector is greater than $\sqrt{3}/2$, no quasi-inversion is applied. An ensuing shift of the real part leads to the matrix

```
W =

  Columns 1 through 3

   0.0409 + 1.3005i   0.0530 + 0.3624i  -0.4849 - 0.6245i
   0.0530 + 0.3624i   0.4364 + 1.0753i  -0.3594 - 0.6598i
  -0.4849 - 0.6245i  -0.3594 - 0.6598i  -0.4706 + 1.3844i
  -0.1064 - 0.4257i   0.1205 - 0.1783i  -0.1946 - 0.1178i
   0.3590 + 0.5023i   0.1990 - 0.1118i  -0.0510 - 0.0073i
  -0.4511 + 0.1383i  -0.0171 + 0.2485i  -0.0543 - 0.3239i
   0.2684 - 0.2975i  -0.4161 + 0.2521i   0.0481 + 0.3949i

  Columns 4 through 6

  -0.1064 - 0.4257i   0.3590 + 0.5023i  -0.4511 + 0.1383i
   0.1205 - 0.1783i   0.1990 - 0.1118i  -0.0171 + 0.2485i
  -0.1946 - 0.1178i  -0.0510 - 0.0073i  -0.0543 - 0.3239i
   0.2815 + 0.7791i   0.3684 - 0.0369i   0.3907 - 0.1531i
   0.3684 - 0.0369i   0.2315 + 0.6895i   0.3656 - 0.1563i
   0.3907 - 0.1531i   0.3656 - 0.1563i  -0.4318 + 0.6585i
  -0.2437 - 0.3094i  -0.2134 - 0.1308i  -0.1541 + 0.0260i

  Columns 7 through 7

   0.2684 - 0.2975i
  -0.4161 + 0.2521i
   0.0481 + 0.3949i
  -0.2437 - 0.3094i
  -0.2134 - 0.1308i
  -0.1541 + 0.0260i
  -0.4997 + 1.0021i.
```

However, the square of the norm of the shortest lattice vector of the imaginary part of the matrix $W$ is 0.6585, well below the threshold $\sqrt{3}/2$. This shows the limitations of the LLL algorithm since the convergence of the theta series is controlled by the length of the shortest lattice vector. Note

that the LLL reduced $\tilde{Y}$ above has the shortest vector in the 6th column (with squared norm 0.6585). One could construct a unimodular matrix $Z$ such that $RZ$ has this vector appearing in the first column (the resulting matrix might not satisfy the LLL condition). This would be more suited to the application of Siegel's algorithm, but will be still approximate since in general LLL does not identify the shortest lattice vector correctly.

Now, if the same algorithm is applied with an exact determination of the shortest vector, the result changes considerably: in the first step of the iteration, the shortest lattice vector is correctly identified having the square of the norm 0.6585. Thus after a shift of the real part, a quasi-inversion is applied. The subsequent identification of the shortest vector of the resulting matrix leads to a vector of squared norm 0.7259. After a shift of the real part, another quasi-inversion is applied. This time the square of the norm of the shortest vector is 1.0211 and thus greater than $\sqrt{3}/2$. After a shift of the real part we finally obtain $W=$

```
Columns 1 through 3

 0.3967 + 1.0211i   0.0615 - 0.1322i  -0.0000 + 0.0000i
 0.0615 - 0.1322i   0.3967 + 1.0211i   0.3553 - 0.5828i
-0.0000 + 0.0000i   0.3553 - 0.5828i   0.2894 + 1.1656i
-0.4609 - 0.2609i  -0.3386 + 0.1933i   0.0905 + 0.2450i
 0.3553 - 0.5828i   0.4776 - 0.1287i  -0.4776 + 0.1287i
 0.1838 + 0.3219i   0.2743 + 0.5669i   0.3871 - 0.3736i
-0.3386 + 0.1933i  -0.3386 + 0.1933i  -0.1223 - 0.4541i


Columns 4 through 6

-0.4609 - 0.2609i   0.3553 - 0.5828i   0.1838 + 0.3219i
-0.3386 + 0.1933i   0.4776 - 0.1287i   0.2743 + 0.5669i
 0.0905 + 0.2450i  -0.4776 + 0.1287i   0.3871 - 0.3736i
 0.3967 + 1.0211i  -0.4776 + 0.1287i   0.0167 - 0.3895i
-0.4776 + 0.1287i   0.2894 + 1.1656i  -0.1671 - 0.7115i
 0.0167 - 0.3895i  -0.1671 - 0.7115i   0.4414 + 1.2784i
 0.0615 - 0.1322i   0.0905 + 0.2450i  -0.3386 + 0.1933i


Columns 7 through 7
```

```
-0.3386 + 0.1933i
-0.3386 + 0.1933i
-0.1223 - 0.4541i
 0.0615 - 0.1322i
 0.0905 + 0.2450i
-0.3386 + 0.1933i
 0.3967 + 1.0211i.
```

In contrast to the algorithm incorporating LLL reductions, the square length of the shortest vector of the imaginary part is here given by the (11) component of the matrix $W$.
Note that the approximate character of the LLL algorithm is unsatisfactory for our purposes for two reasons:
First, the overestimation of the length of the shortest vector leads to a premature end of the algorithm and a much shorter shortest vector than necessary. But secondly, the potentially crude approximation of its length implies that an estimate of the truncation parameter $N_\epsilon$ based on the LLL result could be misleading with the consequence of a loss of accuracy in the approximation of the theta functions.

We have shown in this part that Siegel's reduction can be used to efficiently compute the theta functions and this by replacing the LLL algorithm used by Deconinck et al. by a Minkowski reduction algorithm. In fact, our implementation of Siegel's reduction algorithm allows also a precise estimate of the number in the truncated sum and reduces this number. If for example, we were to gain in dimension $g$ a factor 2 per direction, this would correspond to $2^g$ terms in the sum. hence more $g$ is high, the advantage would be decisive.

## 1.4 Outline of the thesis

The material of this work is arranged as follows:
We start the second chapter by a general introduction about lattices and lattice reduction problems in section 2.1. In section 2.2, we recall the necessary background about lattices (the dual lattice, the Gram-Schmidt orthogonalization, the determinant of a lattice, the complex-valued lattices, the size reduction algorithm, Minkowski's successive minimum and Hermite's constant, Minkowski's theorem), in order to be able to properly define several notions of reduction. In section 2.3 and 2.4, we present the most studied algorithmic problem on Euclidean lattices the SVP and the known algorithms related to lattice reduction especially for the Minkowski reduction and its

different definitions. In section 2.5, we show a particular result due to Tammela in the determination of Minkowski's fundamental domain for genus 3. In chapter 3, we focus on the time complexity of the different reduction algorithms presented in the second chapter where Gauss's reduction and the LLL reduction are the only polynomial reduction algorithms. In chapter 4, we present our Minkowski reduction algorithm up to dimension 5, we start by a description of the algorithm, the drawback of this algorithm, and then we show by examples the differences between these reduction algorithms and the quality of a reduction algorithm measured in terms of the orthogonality defect. Finally, chapter 5 is devoted to study the action of the symplectic group on the Siegel upper half space, especially for genus 1 and 2. A new result is presented for genus 3. This chapter contains also a powerful application of Siegel's fundamental domain to "Riemann's theta function" where the idea was investigated first by Deconinck et al. An efficient computation of this function illustrated by examples is presented in this context by replacing the LLL algorithm used in [38] strategy by our Minkowski reduction algorithm up to dimension 5 and an exact determination of the shortest lattice vector for greater dimensions.

# Chapter 2

# Lattice and lattice reduction

Historically, lattices were investigated since the late of 18th century by Lagrange and Gauss. In the 19th century, important results due to Minkowski motivated the use of lattice theory in the theory and geometry of numbers. The evolution of computer science in the 20th century, especially after the publication of the landmark polynomial time reduction method by Lenstra, Lenstra and Lovász and that was widely known as the LLL algorithm, led to lattice applications in various theoretical areas such as factorization of integers, polynomials, integer programming and public-key cryptography. In the latter area, lattice theory has played a powerful role in the definition of new cryptosystems and in cryptanalysis.

## 2.1 Introduction

A lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^n$. Any lattice can be characterized in terms of a set of $m \leq n$ linearly independent basis vectors $B = \{b_1, ..., b_m\}$ as $\{x : x = \sum_{l=1}^{m} z_l b_l, z_l \in \mathbb{Z}\} = \{Bz : z \in \mathbb{Z}^n\}$ where $m$ is the rank or the dimension of $\mathcal{L}$.
A lattice basis is usually not unique. Given a lattice $\mathcal{L}$ of dimension $m$, $(m \geq 2)$, the lattice $\mathcal{L}$ can have infinitely many different bases, see [77], where two bases are connected by an unimodular matrix $Z$, $B' = BZ$. Some bases are more interesting than others; they are called reduced.
The goal of lattice basis reduction is to find, for a given lattice, a basis matrices with favorable properties. Usually, such a basis consists of vectors that are reasonably short or, equivalently, a basis consisting of vectors that are pairwise nearly orthogonal, is called "a reduced lattice basis".
It is a vague concept and the geometry of numbers has sought from the beginning to clarify this reduced bases concept. This leads to two crucial

questions:

What is the complexity of the main problems related to the reduction? How to construct effective reduction algorithms?

The time complexity of an algorithm quantifies the amount of time taken by an algorithm to run. It is commonly estimated by counting the number of elementary operations performed by the algorithm, where an elementary operation takes a fixed amount of time to perform. An algorithm with $2^{\text{poly}(n)}$ is called an *exponential* time complexity and with $2^{O(\log n)} = \text{poly}(n)$, a *polynomial* time complexity where poly$(n)$ means polynomial in $n$.

There are many lattice reduction algorithms with corresponding reduction criteria and for each of them, there is a notion of quality of the reduced basis and computational complexity required to obtain it: in 1850, Hermite introduced the first notion of reduction, [76]. In 1873, Korkine and Zolotareff, [92], strengthened the definition of Hermite reduction. Their proposed notion is referred to as HKZ reduction, [139]. In 1973, Kannan gave an algorithm for constructing HKZ reduced bases, [87], which in particular contain a shortest non zero lattice element. The algorithm was exponential, but polynomial for fixed dimension. His algorithm yields the asymptotically best known algorithm for integer linear programming as Hermite reduced bases. Minkowski reduced bases contain a shortest nonzero lattice vector, however, Minkowski reduced bases were only constructed for dimensions 2 and 3, [97]. The algorithm was further refined to construct both Minkowski reduced bases and HKZ reduced bases and improved the complexity analysis by Helfrich [74] in 1985, Kannan in 1987 [86], Banihashemi and Khandani [17] in 1998.
Among these, the strongest one is Minkowski reduction. In 1890s, Minkowski [121] defined 'Minkowski reduced' bases, requiring that each basis vector is *as short as possible*. Minkowski dealt with these bases in his *Geometry of Numbers* and in the theory of quadratic forms. Up to dimension four, it is arguably optimal compared to all other known reductions, because it reaches all the so-called *Minkowski's successive minima*, denoted by $\lambda_i$ and defined as the smallest positive number $\lambda$ such that $\lambda\mathcal{L}$ contains at least $i$ linearly independent lattice vectors. However, finding a Minkowski reduced basis cannot be solved in polynomial time under randomized reductions as the dimension increases because such a basis contains a shortest lattice vector and the shortest vector problem cannot be solved in polynomial time under randomized reductions, [125], [128].
Unfortunately, finding good reduced bases has proved invaluable in many fields and the computational complexity of lattice reduction has attracted considerable attention. Therefore, the lattice reduction algorithms yield reduced bases with shorter basis vectors and improved orthogonality. It provided a compromise between the quality of the reduced basis and the computa-

tional effort required for finding it. The lattice reduction algorithms can be grouped into two categories according to their complexity: exponential time algorithms and polynomial time algorithms. There are no known polynomial time algorithm for producing Minkowski reduced bases and HKZ reduced bases [6].

Practical algorithms for computing Minkowski reduced and HKZ reduced lattice bases can be found in [137], for lattices of low dimensions with quadratic complexity, [30], [198], [199], for lattices of arbitrary rank, where [198], [199] used sphere decoding strategies [73] to find a shortest lattice vector inside a sphere centered at 0 of radius $\rho$ and reduced the computational costs of their existing counterparts.

The first polynomial time lattice reduction algorithm was presented in 1982, [103], known as LLL reduced basis, named after the three authors A. Lenstra, W. Lenstra and L. Lovász. Theoretically, the LLL reduction algorithm can produce an approximate shortest vector that is at most a factor of $O(2^m)$ times longer than a real shortest vector of a lattice, [15]. This basis is, of course, not perfect but sufficient for many problems. Further improvements of the LLL algorithm have been developed, while some improve the output quality, [157, 111, 136, 112], others improve the efficiency, [136], [106].
The Jacobi lattice reduction algorithm presented in [146] adopted a different strategy from the LLL algorithm to construct a reduced basis in polynomial time and showed that the Jacobi method outperforms the LLL algorithm in not only efficiency, but also the orthogonality defect (which measures the quality of a such lattice basis). It is equal to 1 if and only if the vectors of a such a lattice basis are orthogonal.

In this chapter, we give the mathematical background on lattices. The chapter mainly includes definitions about lattices, some useful lattice properties and some necessary theorems in section 2.2. Then, we proceed by defining and studying a very important characteristic of a lattice, namely its shortest vector in section 2.3. we describe also various notions of reduction in section 2.4 and present an important result due to Tammela for constructing a Minkowski fundamental domain for dimension 3 in section 2.5.

## 2.2 Lattice

A lattice is a set of points in an $n$-dimensional space with period structure, such as the one illustrated in Figure. 2.1. The theoretical study of lattices is often called the *Geometry of Numbers*, a name bestowed on it by Minkowski in his 1910 book "Geometrie der Zahlen". More recently, lattices have become a

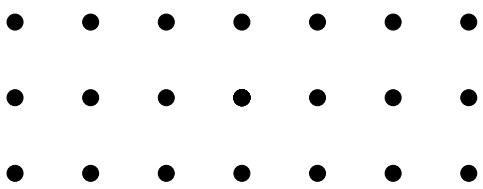

Figure 2.1: Lattice in $\mathbb{R}^2$

topic of active research in computer science. They are used as an algorithmic tool to solve a wide variety of problems as the SVP and the CVP. The practical process of finding short(est) or close(st) vectors in lattices is called "Lattice Reduction".

Let $\mathbb{R}^n$ be the $n$-dimensional Euclidean space. A lattice in $\mathbb{R}^n$ is the set

$$\mathcal{L}(b_1, .., b_m) = \left\{ \sum_{i=1}^{m} x_i b_i ; x_i \in \mathbb{Z} \right\},$$

of all integer combinations of $m$ linearly independent vectors $b_1, b_2, ..., b_m$ in $\mathbb{R}^n$ $(n \geq m)$. The integers $n$ and $m$ are called the dimension and rank of the lattice. The sequence of vectors $b_1, .., b_m$ is called a lattice basis and it is conveniently represented as a matrix

$$B = [b_1, ..., b_m] \in \mathbb{R}^{n \times m},$$

having the basis vectors as columns. Using matrix notation, a lattice can be rewritten as

$$\mathcal{L}(B) = \{Bx : x \in \mathbb{Z}^m\},$$

where $Bx$ is the usual matrix-vector multiplication. If $n = m$, then $\mathcal{L}(B)$ is called a full rank lattice. Note that a full rank lattice will be considered in this context except otherwise indicated.
The basis for a given lattice is not unique. There exist infinitely many bases for a lattice.

**Theorem.** [127] Let $B$ and $\tilde{B}$ be two bases. Then $\mathcal{L}(B) = \mathcal{L}(\tilde{B})$ if and only if there exists a unimodular matrix $U$ such that $B = \tilde{B}U$.

*Proof.* First let $B = \tilde{B}U$ for some unimodular matrix $U$. Notice that if $U$ is unimodular, then $U^{-1}$ is unimodular too. $B = \tilde{B}U$ and $\tilde{B} = BU^{-1}$. Therefore, $\mathcal{L}(B) \subseteq \mathcal{L}(\tilde{B})$ and $\mathcal{L}(\tilde{B}) \subseteq \mathcal{L}(B)$, i.e., the two matrices generate the same lattice.
Now, we assume that $B$ and $\tilde{B}$ are two bases for the same lattice. Then,

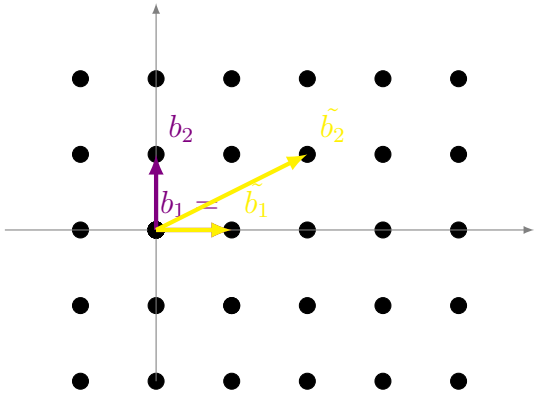


Figure 2.2: A "good" basis and a "bad" basis

by definition of lattices, there exist integer square matrices $V$ and $W$ such that $B = \tilde{B}W$ and $\tilde{B} = BV$. Combining these two equations we obtain, $B = BVW$, or equivalently, $B(I - VW) = 0$. Since the vectors of $B$ are linearly independent, $I - VW = 0$, i.e., $VW = I$ and $V$, $W$ have integer entries, then $\det(V)\det(W) \in \mathbb{Z}$ and $\det(V) = \det(W) = \pm 1$. □

The simplest example is the cubic lattice, obtained by taking the basis vectors $b_i = e_i$ where $e_i$ denotes the $i$th column of the $n$-dimensional identity matrix $I_n$. In this case, we have $B = I_n$ and $\mathcal{L} = \mathbb{Z}^n$. $\mathbb{Z}^n$ can also be generated by the basis vectors $\{e_1, e_2 + ke_1, ..., e_n + ke_1\}$, $k \in \mathbb{Z}$, i.e, $\tilde{B} = B + k(0, e_1, e_1, ..., e_1)$. Since $B = I_n$ is an orthogonal basis, $B$ is the best reduced basis of $\mathcal{L}$. Above, we illustrate the case for $n = 2$, see Figure. 2.2.

### 2.2.1 The Dual Lattice

To any lattice, there is an associated dual lattice, defined by

$$\mathcal{L}^\star = \left\{ x^\star \in \text{Span}\,(B) \mid x^\top x^\star \in \mathbb{Z} \;\text{ for all } x \in \mathcal{L} \right\},$$

where

$$\text{Span}\,(B) = \{Bx \mid x \in \mathbb{R}^m\}.$$

If $B$ is a basis for the primal lattice $\mathcal{L}$, a basis $B^\star$ for the dual lattice $\mathcal{L}^\star$ can be obtained via the right Moore-Penrose pseudo inverse [114], i.e.,

$$D = B^\star = B\left(B^\top B\right)^{-1}.$$

Since $B^\top B^\star = I_m$, it follows that the primal and the dual basis vectors are bi-orthogonal, i.e., $b_l^\top b_k^\star = 0$ for $l \neq k$. Geometrically, this means that the

dual basis vector $b_k^\star$ is orthogonal to the subspace spanned by the primal basis vector $b_1, ..., b_{k-1}, b_{k+1}, ..., b_m$. This is useful since for any $x = Bz \in \mathcal{L}$ we can recover the $k$th integer coefficient via $z_k = x^\top b_k^\star$. The cubic lattice $\mathbb{Z}^n$ is an example of a lattice that is self-dual in the sense that $\mathcal{L} = \mathcal{L}^\star$. Note that, instead of applying a particular lattice reduction algorithm to the basis $B$, a reduction of the dual basis $B^\star$ can be performed. And a reduction of $B^\star$ by a unimodular matrix $T$ corresponds to a reduction of the primal basis $B$ by the unimodular matrix $T^{-1}$.

**Definition.** For an $m$-dimensional lattice spanned by $B$, we call a basis $D \in \mathbb{R}^{n\times m}$ a dual basis if:

1. $\mathrm{Span}(B) = \mathrm{Span}(D)$.
2. $B^\top D = D^\top B = I$.

In other words, $B$ and $D$ are dual bases if they have the same linear span, and any primal and dual basis vector have scalar product $\langle b_i, d_j \rangle = \delta_{ij}$ where

$$\delta_{ij} = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases}$$

It is not hard to check that $D$ is unique for a given $B$, given by $D = B(B^\top B)^{-1}$. $v \in \mathbb{R}^n$ is a dual vector if and only if $B^\top v \in \mathbb{Z}^n$. Therefore, this last condition can be written as $v \in B^{-\top}\mathbb{Z}^n = \mathcal{L}(B^{-\top})$. Thus, $B^{-\top}$is a basis for the dual lattice. We deduce that when $B \in \mathbb{R}^{n\times n}$, the expression for the dual basis given before simplifies to $D = B^{-\top}$.

**Theorem.** If $D$ is the dual basis of $B$ then $\mathcal{L}(B)^\star = \mathcal{L}(D)$.

*Proof.* For any $Dy \in \mathcal{L}(D)$, and all $Bx \in \mathcal{L}(B)$, we have:

$$Dy = B(B^\top B)^{-1} y \in \mathrm{Span}(B).$$

Also, we find that

$$(Dy)^\top (Bx) = y^\top D^\top B x = y^\top x \in \mathbb{Z},$$

which implies that $Dy \in \mathcal{L}(B)^\star$ and $\mathcal{L}(D) \subseteq \mathcal{L}(B)^\star$. Now, we consider an arbitrary vector $v \in \mathcal{L}(B)^\star$, so by the definition of the dual lattice basis, we

obtain that $B^\top v \in \mathbb{Z}^m$ and $v \in \mathrm{Span}(B)$. It follows that $v = Bw$ for some $w \in \mathbb{R}^m$ and

$$v = Bw = B(B^\top B)^{-1}B^\top Bw = D(B^\top v) \in \mathcal{L}(D).$$

This proves $\mathcal{L}(D) \subseteq \mathcal{L}(B)^\star$. □

**Properties of dual lattices.**

**Lemma.** For any lattice $\mathcal{L}$, $(\mathcal{L}^\star)^\star = \mathcal{L}$.

*Proof.* We assume that $B$ is the basis of $\mathcal{L}$

$$B(B^\top B)^{-1}((B(B^\top B)^{-1})^\top B(B^\top B)^{-1})^{-1} = B(B^\top B)^{-1}((B^\top B)^{-1})^{-1} = B.$$

□

**Lemma.** For a lattice $\mathcal{L}$,

$$\det(\mathcal{L}^\star) = \frac{1}{\det(\mathcal{L})}.$$

*Proof.* Let $B$ be the basis of $\mathcal{L}$. Then, we have

$$\begin{aligned}
\det(\mathcal{L}^\star) &= \sqrt{(B(B^\top B)^{-1})^\top B(B^\top B)^{-1}} \\
&= \sqrt{(B^\top B)^{-1}(B^\top B)(B^\top B)^{-1}} \\
&= \sqrt{(B^\top B)^{-1}} \\
&= \frac{1}{\det(\mathcal{L})}.
\end{aligned}$$

□

**Lemma.** For any lattice $\mathcal{L}$ of rank $m$,

$$\lambda_1(\mathcal{L})\lambda_1(\mathcal{L}^\star) \leq m,$$

where $\lambda_1$ is the first successive minimum.

*Proof.* From Minkowski's bound, [126], (see also 2.2.7), we have

$$\lambda_1(\mathcal{L}) \leq \sqrt{m}\det(\mathcal{L})^{\frac{1}{m}}.$$

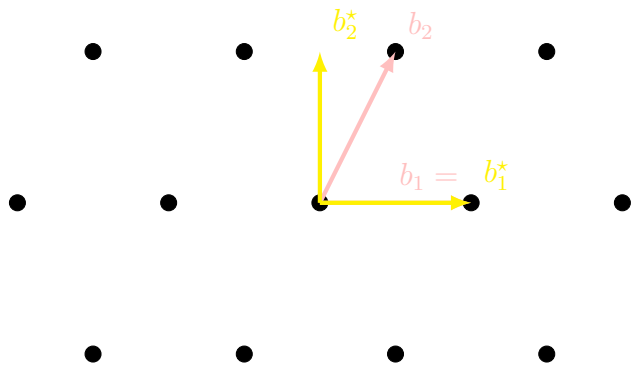


Figure 2.3: Gram-Schmidt orthogonalization

Since $\text{Span}(\mathcal{L}) = \text{Span}(\mathcal{L}^\star)$,

$$\lambda_1(\mathcal{L}^\star) \leq \sqrt{m}\det(\mathcal{L}^\star)^{\frac{1}{m}}.$$

Multiplying and using the previous lemma, we obtain the desired result. □

**Lemma.** For a lattice $\mathcal{L}$,

$$\lambda_1(\mathcal{L})\lambda_m(\mathcal{L}^\star) \geq 1.$$

*Proof.* Let $v \in \mathcal{L}$ such that $\lambda_1 = \|v\|$, $v$ is a vector that achieves the first successive minimum of the lattice $\mathcal{L}$. Take any set $x_1, ..., x_m$ of $m$ linearly independent vectors in $\mathcal{L}^\star$. Not all of them are orthogonal to $v$. Hence, there exists an $i$ such that $\langle x_i, v\rangle \neq 0$. By the definition of the dual lattice, we have $\langle x_i, v\rangle \in \mathbb{Z}$ and hence

$$\lambda_1(\mathcal{L})\lambda_m(\mathcal{L}^\star) \geq \|v\| \,.\, \|x_i\| \geq |\langle v, x_i\rangle| \geq 1.$$

□

More generally, for any $i$,

$$\lambda_i(\mathcal{L})\lambda_{m-i+1}(\mathcal{L}^\star) \geq 1.$$

### 2.2.2 Gram-Schmidt Orthogonalization

Any basis $B$ can be transformed to an orthogonal basis for the same vector space using the well-known Gram-Schmidt orthogonalization method. Suppose

we have vectors $B = [b_1, .., b_m] \in \mathbb{R}^{n \times m}$ generating a vector space $V = \text{Span}\,(B)$. These vectors are not necessarily orthogonal, but we can always find an orthogonal basis $B^* = [b_1^*, ..., b_m^*]$ for $V$ where $b_i^*$ is the component of $b_i$ orthogonal to $\text{Span}\,(b_1, ..., b_{i-1})$.

**Definition.** For any sequence of vectors $B = [b_1, .., b_m]$, we define the orthogonalized vectors $B^* = [b_1^*, ..., b_m^*]$ iteratively according to the formula

$$b_i^* = b_i - \sum_{j<i} \mu_{i,j} b_j^* \text{ where } \mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}. \qquad (\star)$$

In matrix notation, $B = B^* M$ where $M$ is the upper triangular matrix with 1 along the diagonal and $m_{i,j} = \mu_{i,j}$ for all $j < i$. It also follows that $B^* = BM^{-1}$ where $M^{-1}$ is also upper triangular with 1 along the diagonal. Note that the columns of $B^*$ are orthogonal $\left(\langle b_i^*, b_j^* \rangle = 0 \text{ for all } i \neq j\right)$. Therefore, the (non-zero) columns of $B^*$ are linearly independent and form a basis for the vector space $\text{Span}\,(B)$. However they are generally not a basis for the lattice $\mathcal{L}\,(B)$.

**Example.** The Gram-Schmidt orthogonalization of the basis $B = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$ is $B^* = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$. However this is not a lattice basis for $\mathcal{L}\,(B)$ because $\begin{pmatrix} 0 \\ 2 \end{pmatrix}$ does not belong to the lattice. $\mathcal{L}\,(B)$ contains a sublattice generated by a pair of orthogonal vectors $\begin{pmatrix} 2 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 4 \end{pmatrix}$ but no pair of orthogonal vectors generate the entire lattice $\mathcal{L}\,(B)$.

So, while vector spaces always admit an orthogonal basis, this is not true for lattices. Let $B = [b_1, .., b_m]$ be a basis for a lattice $\mathcal{L}$ in $\mathbb{R}^n$ and $B^* = [b_1^*, ..., b_m^*]$ be its Gram-Schmidt orthogonalization. For $i \in \{1, ..., m\}$, denote by $\pi_i : \mathbb{R}^n \to (\mathbb{R}b_1 + ... + \mathbb{R}b_{i-1})^\perp$ the orthogonal projection on the orthogonal complement of $\mathbb{R}b_1 + .. + \mathbb{R}b_{i-1}$. $\mathcal{L}^{m-i+1} = \pi_i(\mathcal{L})$, this is a lattice of rank $m - i + 1$ with basis $[\pi_i(b_i), ..., \pi_i(b_m)]$. In terms of the Gram-Schmidt decomposition we have

$$\pi_i(b_j) = b_j^* + \sum_{k=i}^{j-1} \mu_{jk} b_k^*,$$

In particular, $\pi_i(b_i) = b_i^*$.

What is the dual of $\pi_i(\mathcal{L}(B))$? It is the sublattice of $\mathcal{L}(D)$ generated by $d_i, ..., d_m$

**Lemma.** Let $B, D \in \mathbb{R}^{n \times m}$ be a pair of dual bases. For all $i = 1, ..., m$, $[\pi_i(b_i), .., \pi_i(b_m)]$ and $[d_i, ..., d_m]$ are also dual bases.

*Proof.* We only prove the statement for $i = 2$. The general statement follows easily by induction on $i$. Therefore, let $B' = [\pi_2(b_2), ..., \pi_2(b_m)]$ and $D' = [d_2, .., d_m]$. So, we have to verify first that $B'$ and $D'$ span the same vector space and secondly that $(B')^\top (D') = I$.
We start with the second one. For all $i, j > 1$, we have

$$\begin{aligned} \langle \pi_2(b_i), d_j \rangle &= \langle b_i - \mu_{i1} b_1, d_j \rangle \\ &= \langle b_i, d_j \rangle - \mu_{i1} \langle b_1, d_j \rangle \\ &= \delta_{ij} - \mu_{i1} \delta_{1j} \\ &= \delta_{ij} \end{aligned}$$

This proves that $(B')^\top (D') = I$.
For the first one, we know that $B$ and $D$ span the same vector space $V$. The linear span of $B'$ is by definition the orthogonal complement of $b_1$ in $V$. Since the vectors $d_2, ..., d_m$ are all orthogonal to $b_1$ (by definition of dual basis) and they are linearly independent, they also span the orthogonal complement of $b_1$ in $V$. So, $B'$ and $D'$ have the same linear span and this proves the lemma. □

In contrast, another orthogonalization approach is the **QR** decomposition.

**Definition.** The **QR** decomposition is a decomposition obtained by applying a sequence of Householder or Givens transformations, that factorizes $\mathbf{B}$ according to $\mathbf{B} = \mathbf{QR}$, where $\mathbf{Q} = (q_1, ..., q_m)$ is an $n \times m$ columns-orthogonal matrix, we have $\mathbf{Q}^\top \mathbf{Q} = I_m$, and $\mathbf{R}$ is an $m \times m$ upper triangular matrix with positive diagonal element.

The **QR** decomposition amounts to expressing the $l$th column of $\mathbf{B}$ in terms of the orthonormal basis vectors $q_1, ..., q_l$ as

$$b_l = \sum_{k=1}^{l} r_{k,l} q_k.$$

Here, $q_k^\top b_l = r_{k,l}$ characterizes the component of $b_l$ collinear with $q_k$. Furthermore, $r_{l,l}$ describes the component of $b_l$ which is orthogonal to the space

spanned by $b_1, .., b_{l-1}$, or, equivalently, by $q_1, ..., q_{l-1}$. A basis vector $b_l$ is almost orthogonal to the space spanned by $b_1, .., b_{l-1}$, if the absolute values of $r_{1,l}, ..., r_{l-1,l}$ are close to zero. If these elements of $\mathbf{R}$ are exactly zero, $b_l$ has no component in the direction of $b_1, .., b_{l-1}$ and is correspondingly orthogonal to the space spanned by these vectors. However, for general lattices such a strictly orthogonal basis does not exist and one has to settle for a basis satisfying less stringent criteria. Instead of GSO, many recent lattice reduction algorithms [195], [110], [193],[194], [131], [29] adopt the **QR** decomposition approach, since the **QR** decomposition can be performed more efficiently than the Gram-Schmidt orthogonalization. We have the following relations between **QR** decomposition and the Gram-Schmidt orthogonalization:

- $q_l = \frac{b_l^*}{\|b_l^*\|}$.
- $\|b_l^*\| = r_{l,l}$.
- $\mu_{l,k} = \frac{r_{k,l}}{r_{k,k}}$.

Another practical method will be used in this context, and it is called the "**Cholesky factorization**".

**Definition.** A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is positive definite if

$$x^\top Ax > 0 \text{ for all } x \neq 0$$

Note that if $A$ is an $n \times n$ symmetric matrix, then $x^\top Ax$ is the function

$$x^\top Ax = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j = \sum_{i=1}^{n} A_{ii} x_i^2 + 2 \sum_{i>j} A_{ij} x_i x_j$$

this is called "a quadratic form".

**Definition** (Cholesky Factorization)**.** Every positive definite matrix $A \in \mathbb{R}^{n \times n}$ can be factored as

$$A = R^\top R$$

where $R$ is upper triangular matrix with positive diagonal elements.

We define now a fundamental quantity associated to any lattice, *the determinant*.

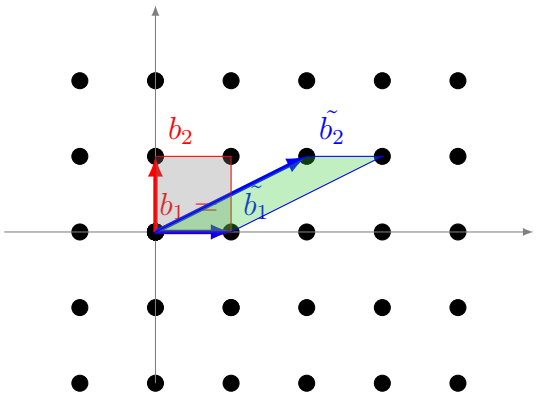


Figure 2.4: Example lattice $\mathbb{Z}^2$ with bases $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\tilde{B} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$ and associated fundamental parallelograms

### 2.2.3 The Determinant

Given a basis $B = [b_1, ..., b_m] \in \mathbb{R}^{n \times m}$, the fundamental parallelepiped associated to $B$ is the set of points

$$\mathcal{P}(B) = B[0,1)^m = \left\{ \sum_{i=1}^{m} x_i . b_i : 0 \leq x_i < 1 \right\}.$$

$\mathcal{P}(B)$ is also a so-called fundamental region, i.e, a region that completely covers the span of $B$ when shifted to all points of the lattice (for any $x \in \mathbb{R}^n$, there exists a unique lattice point $v \in \mathcal{L}(B)$, such that $x \in (v + \mathcal{P}(B))$. Another important fundamental region is the **Voronoi region**, defined as the set of points in $\mathbb{R}^n$ that are closer to the origin than to any other lattice point,

$$\mathcal{V}(\mathcal{L}) := \{x | \, \|x\| \leq \|x - y\| \text{ for all } y \in \mathcal{L}\}.$$

In contrast to the fundamental parallelepiped $\mathcal{P}(B)$, the Voronoi region $\mathcal{V}(\mathcal{L})$ is a lattice invariant, i.e., it is independent of the specific choice of a lattice basis. However, the volume (here, volume is defined in the $m$-dimensional space spanned by the columns of $B$) of $\mathcal{P}(B)$ is the same for all bases of a given lattice. This volume equals the so-called lattice determinant, which is a lattice invariant defined as the square-root of the determinant of the Gramian $B^\top B$,

$$|\mathcal{L}| := \text{Vol}(\mathcal{P}(B)) = \prod_i \|b_i^*\| = \sqrt{\det(B^\top B)},$$

where $B^*$ is the Gram-Schmidt orthogonalization of $B$. The first part is the definition of the determinant of a lattice which is a generalization of the well

known formula for the area of a parallelepiped. A useful application of the Gram-Schmidt process is the following:
Let $b_1, ..., b_m$ be a set of m linearly independent vectors in $\mathbb{R}^n$ and consider the orthonormal basis vectors given by $b_1^*/\left\|b_1^*\right\|, ...., b_m^*/\left\|b_m^*\right\|$. In this basis, the vectors $b_1, ..., b_m$ are given as the columns of the $n \times m$ matrix

$$\begin{bmatrix} \left\|b_1^*\right\| & \mu_{21}\left\|b_1^*\right\| & \cdots & & \mu_{m1}\left\|b_1^*\right\| \\ 0 & \left\|b_2^*\right\| & \cdots & & \mu_{m2}\left\|b_2^*\right\| \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & \left\|b_{m-1}^*\right\| & \mu_{mm-1}\left\|b_{m-1}^*\right\| \\ 0 & \cdots & & 0 & \left\|b_m^*\right\| \\ 0 & \cdots & & 0 & 0 \\ \vdots & \cdots & & \vdots & \vdots \\ 0 & \cdots & & 0 & 0 \end{bmatrix}$$

In the case $n = m$ this is an upper-triangular square matrix. From this representation, it is easy to see that the volume of $\mathcal{P}(b_1, ..., b_n)$, or equivalently, $\det(\mathcal{L}(b_1, .., b_n))$, is given by $\prod_{i=1}^n \left\|b_i^\star\right\|$.
If the lattice has full rank, the lattice determinant equals the magnitude of the determinant of the basis matrix $B$, i.e.,

$$|\mathcal{L}| = |\det(B)|.$$

Remember the Gram-Schmidt orthogonalization procedure $(\star)$. In matrix notation, it shows that the orthogonalized vectors $B^*$ satisfy $B = B^*T$, where $T$ is an upper triangular matrix with 1's on the diagonal, and the $\mu_{i,j}$ coefficients at position $(j, i)$ for all $j < i$.So, our formula for the determinant of a lattice can be written as

$$\sqrt{\det(B^\top B)} = \sqrt{\det(T^\top B^{*\top} B^* T)} = \sqrt{\det(T^\top)\det(B^{*\top} B^*)\det(T)}.$$

The matrices $T^*$, $T$ are triangular, and its determinant can be easily computed as the product of the diagonal elements, which is 1. Now consider $B^{*\top} B^*$. This matrix is diagonal because the columns of $B^*$ are orthogonal. So, its determinant can also be computed as the product of the diagonal elements which is

$$\det\left(B^{*\top} B^*\right) = \prod_i \langle b_i^*, b_i^* \rangle = \left(\prod_i \left\|b_i^*\right\|\right)^2 = \det(\mathcal{L}(B))^2.$$

Taking the square root we get

$$\sqrt{\det(T^\top)\det(B^{*\top} B^*)\det(T)} = \det(\mathcal{L}(B)).$$

### 2.2.4 Complex-Valued Lattices

A complex-valued lattice of rank $m$ in the $n$-dimension complex space $\mathbb{C}^n$ is defined as

$$\mathcal{L} = \left\{ x \mid x = \sum_{l=1}^{m} z_l b_l, \quad z_l \in \mathbb{Z}_j \right\},$$

with complex basis vectors $b_l \in \mathbb{C}^n$ and $\mathbb{Z}_j = \mathbb{Z} + j\mathbb{Z}$ denoting the set of complex integers (also known as Gaussian integers). By arranging the basis vectors into an $n \times m$ complex-valued matrix $B$ and noticing that the complex mapping $x = Bz$ can be equivalently expressed as

$$x = \begin{pmatrix} \mathcal{R}(x) \\ \mathcal{I}(x) \end{pmatrix} = \begin{pmatrix} \mathcal{R}(B) & -\mathcal{I}(B) \\ \mathcal{I}(B) & \mathcal{R}(B) \end{pmatrix} \begin{pmatrix} \mathcal{R}(z) \\ \mathcal{I}(z) \end{pmatrix},$$

In fact, since a $m$-dimensional complex lattice is isomorphic to a $2m$-dimensional real lattice, every decoding problem that has a complex lattice formulation can also be reformulated as a real lattice decoding problem. Many of the concepts and algorithms from the real-valued space can be formulated directly in the complex domain with minor modifications [113], [161], [162], [57], [132], [58], [107].

For example, let $H = \{h_1, ..., h_m\}$ an $n \times m$ complex matrix, and the conjugate transpose (Hermitian) of a matrix $H$ is denoted by $H^H$. The inner product of two vectors $h_1$ and $h_2$ is defined as $\langle h_1, h_2 \rangle = h_2^H h_1$. The set of orthogonal vectors generated by the Gram-Schmidt Orthogonalization procedure are represented as $\{h_1^*, ..., h_m^*\}$ which span the same space as $\{h_1, ..., h_m\}$, and further

$$\mu_{ij} = \frac{\left\langle h_i, h_j^* \right\rangle}{\left\| h_j^* \right\|^2},$$

Note that working directly on the complex lattice can result in decoding algorithms with lower complexity, because the exploitation of the complex lattice structure allows the lattice dimension involved to be only half of that of the equivalent real lattice.

**Definition** (The orthogonality Defect.)**.** The "quality" of a lattice basis can be measured in terms of the orthogonality defect, defined as

$$\delta(B) = \frac{1}{\det(\mathcal{L}(B))} \prod_{l=1}^{m} \|b_l\|.$$

For any $m \times m$ positive definite matrix $A$ with elements $a_{k,l}$, the Hadamard inequality, [67], states that $\det(A) \leq \prod_{l=1}^{m} a_{l,l}$ with equality if and only if $A$

is diagonal. Setting $A = B^\top B$ this implies that the orthogonality defect is bounded from below as $\delta(B) \geq 1$, with equality if and only if $B$ is orthogonal. Reduce a lattice basis means transform this lattice basis to an orthogonal one or constitute a basis where the lengths of the bases vector are close to successive minima. This is why some reductions are trying to imitate the orthogonalization of Gram-Schmidt.

### 2.2.5 Size Reduction

A rather simple but not very powerful criterion is given by the so-called "Size Reduction", often called weakly reduced. A basis $B$ is size-reduced if the elements of the corresponding upper triangular matrix $R$ satisfy the following condition:

$$|r_{kl}| \leq \frac{1}{2}|r_{kk}| \quad \text{for } 1 \leq k < l \leq m.$$

We can also say, a basis is called size-reduced if

$$|\mu_{kl}| \leq \frac{1}{2}, \quad \text{for } 1 \leq i < j \leq m \quad \text{where} \quad \mu_{kl} = \frac{\langle b_k, b_l \rangle}{\langle b_l, b_l \rangle}.$$

### 2.2.6 Minkowski's Successive Minimum And Hermite's Constant

Let $\mathcal{L}$ be an $m$-dim lattice in $\mathbb{R}^n$. For $1 \leq i \leq m$, the $i$th Minkowski's successive minimum, $\lambda_i(\mathcal{L})$, is the radius of the smallest closed ball centered at the origin containing at least $i$ linearly independent lattice vectors.

$$\lambda_i(\mathcal{L}) = \min\left(r, \dim(\mathcal{B}(0,r) \cap \mathcal{L}) \geq i\right).$$

In particular, $\lambda_1(\mathcal{L})$ is the Euclidean length of the shortest non zero lattice vector of $\mathcal{L}$. In dimension 2, we say that a basis $(b_1, b_2)$ of a lattice $\mathcal{L}$ is reduced, if it takes out the first and the second minimum.

**Proposition.** The successive minima of a lattice are always reached, there always exist independent lattice vectors $v_i$'s such that

$$\|v_i\|_2 = \lambda_i(L), \text{ for all } i$$

*Proof.* We consider an $n$-ball, $B_n$, centered at the origin with radius $r$. $r$ is large enough such that $B_n$ contains a non zero lattice vector. $D = (\mathcal{L} \cap B_n) - \{0\}$ is non-empty. Then, we restrict $\|.\|$ to $D$: $\|.\| : D \to \mathbb{R}$. Since $D$

contains a finite number of lattice points, it is compact and $\|.\|$ is continuous, $\|D\| = \{\|d\| : d \in D\} \subset \mathbb{R}$ is also compact. Since every compact subset of $\mathbb{R}$ contains a smallest and a largest element, $\|D\|$ contains a smallest say $b$. By the property of norm, $b \geq 0$. Since $D$ contains only non zero vectors, $b > 0$. Therefore, there exists a non zero lattice vector of minimal length in $D$. $\square$

Note that for $m > 4$, such vectors do not necessarily form a basis for $\mathcal{L}$. Let us consider for example the following lattice, in dimension 5:

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

This lattice contains the vector below:

$$V = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

It is easy to notice that the successive minima are:

$$\lambda_1 = 2,\ \lambda_2 = 2,\ \lambda_3 = 2,\ \lambda_4 = 2,\ \lambda_5 = 2.$$

However, a family realizing these minimum, does not contain necessarily $V$, as for example:

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Now, we are going to give Minkowski's theorem, which allows to finitely increase the minimum of a given lattice and to make the link between a simple invariant to calculate, the volume, and the successive minima.

### 2.2.7 Minkowski's Theorem

In this section, we consider the classical result of Minkowski, which, in some sense, originated the whole geometry of numbers.

**Definition.** A set $S \subset \mathbb{R}^n$ is said to be centrally-symmetric about the origin, if $x \in S$ implies $-x \in S$. The set $S$ is said to be convex, if for any $x, y \in S$, $tx + (1-t)y \in S$ for $0 \leq t \leq 1$: that is $S$ contains the line segment that passes through $x$ and $y$.

**Theorem** (Blichfeldt, [21])**.** For any full rank lattice $\mathcal{L} \subset \mathbb{R}^n$ and a measurable set $S \subset \mathbb{R}^n$ with $\text{vol}(S) > \det(\mathcal{L})$, there exist distinct vectors $z_1, z_2 \in S$ such that $z_1 - z_2 \in \mathcal{L}$.

*Proof.* Let $B$ be a basis of $\mathcal{L}$. $x$ ranges over all $\mathcal{L}$, therefore, $x + \mathcal{P}(B) := \{x + y : y \in \mathcal{P}(B)\}$ form a partition of $\mathbb{R}^n$, [166]. Let us define $S_x = S \cap (x + \mathcal{P}(B))$. Since $S = \bigcup_{x \in \mathcal{L}} S_x$, we deduce that $\text{vol}(S) = \sum_{x \in \mathcal{L}} \text{vol}(S_x)$. Now, consider the translates $\hat{S}_x = S_x - x$. Clearly, $\hat{S}_x \subseteq \mathcal{P}(B)$. Now, we argue that these translates $\hat{S}_x$ cannot be pairwise disjoint. Indeed, since $\text{vol}(\hat{S}_x) = \text{vol}(S_x)$, the total volume of all these sets is

$$\sum_{x \in \mathcal{L}} \text{vol}(\hat{S}_x) = \sum_{x \in \mathcal{L}} \text{vol}(S_x) = vol(S) > vol(\mathcal{P}).$$

Therefore, there must exist some $x, y \in \mathcal{L}$, $x \neq y$ for which $\hat{S}_x \cap \hat{S}_y \neq \emptyset$. Let $z$ be a point in $\hat{S}_x \cap \hat{S}_y$. Then, $z + x \in S_x \subseteq S$, $z + y \in S_y \subseteq S$, and $(z + x) - (z + y) = x - y \in \mathcal{L}$. □

We will give an upper bounds on the length of the shortest vector in a lattice. For this, we start by giving and proving a very important theorem (Minkowski's convex body theorem).

**Theorem 3** (Minkowski's Convex Body Theorem)**.** Let $\mathcal{L}$ be a full rank lattice of rank $m$. Then, for any centrally-symmetric convex set $S$, if $\text{vol}(S) > 2^m \det(\mathcal{L})$ then $S$ contains a non-zero lattice point.

*Proof.* Let us define $\hat{S} = \frac{1}{2}S = \{x : 2x \in S\}$. Then, $\text{vol}(\hat{S}) = 2^{-m}\text{vol}(S) > \det(\mathcal{L})$. By Blichfeldt's theorem, there exist two points $z_1, z_2 \in \hat{S}$ such that $z_1 - z_2 \in \mathcal{L}$ is a non zero lattice point. By definition, $2z_1, 2z_2 \in S$ and because is centrally-symmetric, also $-2z_2 \in S$. Finally, since $S$ is convex, $\frac{2z_1 - 2z_2}{2} = z_1 - z_2$ is in $S$. □

**Theorem** (**The first theorem of Minkowski**)**.** Let $\mathcal{L}$ be a lattice of dimension $m$. Then, we have:

$$\lambda_1(\mathcal{L}) \leq \sqrt{m}\ \text{vol}\,(\mathcal{L})^{\frac{1}{m}}\,.$$

*Proof.* We first bound the volume of the ball $\mathcal{B}(0,r)$, for some radius $r$. This ball contains the hypercube $\left[-\frac{r}{\sqrt{m}}, \frac{r}{\sqrt{m}}\right]^m$. Hence, its volume is greater than $\left(\frac{2r}{\sqrt{m}}\right)^m$.
For $r = \sqrt{m}\det(\mathcal{L})^{\frac{1}{m}}$, the volume of $\mathcal{B}(0,r)$ is greater than $2^m \det(\mathcal{L})$, so by Minkowski's convex body theorem (see, theorem. 3), the ball contains a non zero lattice vector, and therefore, the length of the shortest vector is at most $\sqrt{m}\det(\mathcal{L})^{\frac{1}{m}}$. □

When the dimension remains small, we know explicitly the constant of Hermite, [24]. It is the supremum of the following quantities as $\mathcal{L}$ ranges over all $m$-dimensional lattices:

$$\gamma_m = \max_{\mathcal{L},\ \dim\mathcal{L}=m} \left(\frac{\lambda_1(\mathcal{L})}{\mathrm{vol}(\mathcal{L})^{\frac{1}{m}}}\right)^2 \leq m,$$

We notice that Hermite's lattice constant is used to scale the size of the first minimum $\lambda_1(\mathcal{L})$. However, finding the exact value of $\gamma_m$ is a very difficult problem, which plays a central role in the theory of geometry of numbers. The exact value of $\gamma_m$ is only known for $1 \leq m \leq 8$ and $m = 24$, [139]. An upper bound of Hermite's constant is given in [139]:

$$\gamma_m \leq 1 + \frac{m}{4}, \text{ for all } m \geq 1.$$

In the following table, we give the values known to date.

| m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 24 |
|---|---|---|---|---|---|---|---|---|---|
| $(\gamma_m)^m$ | 1 | $\frac{4}{3}$ | 2 | 4 | 8 | $\frac{64}{3}$ | 64 | $2^8$ | $4^{24}$ |

It has never been proved that $\gamma_m$ is an increasing function of $m$. For convenience we define

$$\gamma_m^\star = \max\{\gamma_i : 1 \leq i \leq m\}$$

to obtain a non-decreasing function of $m$.

The first theorem of Minkowski admits a stronger version, which allows to limit the product of the successive minimum. The above theorem easily generalizes to other minima.

**Theorem** (The second theorem of Minkowski.)**.** Let $\mathcal{L}$ be a lattice of dimension $m$, then we have:

$$\prod_{i=1}^{m} \lambda_i(\mathcal{L}) \leq \gamma_m^m \mathrm{vol}(\mathcal{L}) \leq m^{\frac{m}{2}} \mathrm{vol}(\mathcal{L}).$$

Where $\lambda_i(\mathcal{L})$ is the length of the $i$th shortest vector.

**Proposition.** For every lattice basis $B$ and its Gram-Schmidt orthogonalization $B^*$,

$$\lambda_j(\mathcal{L}(B)) \geq \min_{1 \leq i \leq m} \|b_i^*\| . \tag{2.1}$$

*Proof.* Note that $b_i^*$ are not lattice vectors. Let us consider a generic lattice vector

$$Bx \in \mathcal{L}(B) \setminus \{0\} , \tag{2.2}$$

where $x \in \mathbb{Z}^m \setminus \{0\}$ and let $k$ be the biggest index such that $x_k \neq 0$. We prove that

$$\|Bx\| \geq \|b_k^*\| \geq \min_i \|b_i^*\| . \tag{2.3}$$

In order to prove (2.3), we take the scalar product of our lattice vector and $b_k^*$. Using the orthogonality of $b_k^*$ and $b_i$ (for $i < k$) we get

$$\langle Bx, b_k^* \rangle = \sum_{i \leq k} \langle b_i x_i, b_k^* \rangle = x_k \langle b_k, b_k^* \rangle = x_k \|b_k^*\|^2 . \tag{2.4}$$

By Cauchy-Schwarz inequality, i.e., $|\langle x, y \rangle| \leq \|x\| \, \|y\|$,

$$\|Bx\| . \|b_k^*\| \geq |\langle Bx, b_k^* \rangle| \geq |x_k| . \|b_k^*\|^2 . \tag{2.5}$$

Using $|x_k| \geq 1$ and dividing by $\|b_k^*\|$, we get $\|Bx\| \geq \|b_k^*\|$. □

## 2.3 The Shortest vector problem and Sphere decoding algorithms

Minkowski's first theorem gives a simple way to bound the length $\lambda_1$ of the shortest non zero vector in a lattice $\mathcal{L}(B)$. In general $\lambda_1$ can be smaller than $\sqrt{m} \det(B)^{\frac{1}{m}}$ where $\mathcal{L}(B)$ is a full rank lattice.

For example, consider the two-dimensional lattice generated by orthogonal vectors $b_1 = \epsilon e_1$ and $b_2 = \left(\frac{1}{\epsilon}\right) e_2$. The determinant of the lattice is 1 and $\lambda_1 \leq \sqrt{2}$. However $\lambda_1 = \epsilon$ can be arbitrarily small.

Moreover, we know from Minkowski's theorem (see, Section. 2.2.7) that a shortest non zero vector exists, but it doesn't give any computational method to efficiently find vectors of length bounded by $\sqrt{m} \det(\mathcal{L})^{\frac{1}{m}}$, let alone vectors of length $\lambda_1$. The problem of finding a lattice vector of length $\lambda_1$ is the well-known, shortest vector problem (SVP).

**Definition** (Shortest Vector Problem, SVP)**.** Given a basis $B \in \mathbb{R}^{n \times m}$, find a non zero lattice vector $Bx$ (with $x \in \mathbb{Z}^m \setminus \{0\}$) such that $\|Bx\| \leq \|By\|$ for any other $y \in \mathbb{Z}^m \setminus \{0\}$.

The lack of efficient algorithms to solve SVP has led computer scientists to consider approximative versions of the problem.

### 2.3.1 The Closest Point And The Shortest Vector Problem

The Closest-Point problem is the problem of finding for a given lattice $\mathcal{L}$ and a given input point $x \in \mathbb{R}^n$, a vector $\hat{c} \in \mathcal{L}$ such that

$$\|x - \hat{c}\| \leq \|x - c\| \text{ for all } c \in \mathcal{L}, \tag{2.6}$$

where $\|.\|$ denotes the Euclidean norm. The shortest vector problem is a special case of the closest point problem, the idea is to use $x = 0$ as the input and exclude 0 as a potential output,

$$\min_{z \neq 0} \{\|\mathbf{B}z\|_2 \mid z = \lceil z_i \rfloor \in \mathbb{Z}^m\}, \tag{2.7}$$

where $\mathbf{B}$ is the lattice basis of $\mathcal{L}$.

As already said, the shortest vector problem is to find a vector in $\mathcal{L} \setminus \{0\}$ that has the smallest Euclidean norm. Thus, the history of the shortest vector problem is closely linked with that of the closest point problem.

This problem was studied in a very intense way for more than 25 years. It has been conjectured [188] that the shortest vector problem (with $\mathcal{L} \subseteq \mathbb{Z}^n$) is NP-hard, in other words, cannot be solved in polynomial time. But, in contrast to the closest point problem, this is still not proved. The conjecture of [188] is supported by the result of Ajtai [7] who showed that the shortest vector problem cannot be solved in polynomial time under randomized reductions. Micciancio [125] furthermore proved that finding an approximate solution within any constant factor less than $\sqrt{2}$ cannot be also solved in polynomial time for randomized reductions. It is known [62], [75], however, that the shortest vector problem is not harder than the closest vector problem.

Therefore, the answer to the question: "What is the best and fastest algorithm available to determine the shortest vector of a lattice?" is not immediately clear. The choice of a method for solving the shortest vector problem depends on the structure of the lattice. For many classical lattices, efficient search methods are known [35], [189]. Here, we address the problem for general lattices.

The first SVP algorithm was Gauss's reduction algorithm (details in Section. 2.4), which solves SVP exactly in dimension 2, in quadratic time. In arbitrary dimension, there are two types of SVP algorithms:

**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. Exact algorithms can be split into two categories:

**a) Polynomial-Space exact algorithms** In general, the common feature of these deterministic algorithms is to first identify a region in which a shortest lattice point must lie, and then exhaustively search the lattice points lying in this region for the shortest non zero lattice vector, while possibly reducing the size of the region dynamically. They are based on enumeration which dates back to the early 1980s with work by Pohst [142], Fincke-Pohst [49], and named sphere decoding algorithms, [5], [159], [142], [49], [191], in which they examined lattice points lying inside a hypershere. Also, we have algorithms based on Kannan's strategy, [87], [74], [86], [17]. Let $\{b_1, .., b_m\}$ be a basis for an $m$-dim lattice $\mathcal{L}$ and let $\pi_2(.)$ be an orthogonal projection operator which projects "." onto $b_1^*$, where $b_1^*$ denotes the orthogonal complement of the subspace spanned by $b_1$. Note that a basis $\{b_1, .., b_m\}$ of a lattice $\mathcal{L}$ is called a **HKZ** basis if and only if $\{b_1, .., b_m\}$ is weakly reduced and $b_i^*$ is a shortest non zero vector in $\mathcal{L}^{m-i+1}$, $i = 1, .., m$ (see details in Section. 2.4). The basic idea of Kannan enumeration is to first find an **HKZ reduced basis** for the lattice $\pi_2(\mathcal{L})$ by calling itself recursively, and then lift it to a size-reduced basis $\{b_1, b_2', ..., b_m'\}$ for $\mathcal{L}$ such that $\{\pi_2(b_2), ..., \pi_2(b_m')\}$ is an HKZ-reduced basis of $\pi_2(\mathcal{L})$ and $\|b_1\|_2^2 \leq \frac{4}{3}\|b_2'\|_2^2$. Then the shortest lattice point must lie in a parallelepiped of cardinality no more than $m^{0.5m+O(1)}$ and can thus be found by enumerating this finite set (see, [84]). It is proved in [74], [86] that algorithms based on Kannan's strategy require a complexity of $m^{0.5m+O(m)}$ polynomial-time operations. Variants of Kannan's [87], [74], [86], [17] differ mainly in how the size of the search region for each iteration level are chosen.

**b) Exponential-Space exact algorithm** These algorithms have a better asymptotic running time, but they all require exponential space $2^{O(m)}$. The first algorithm of this kind is the randomized sieve algorithm of Ajtai, Kumar and Sivakumar (AKS), [9], with exponential worst-case complexity of $2^{O(m)}$ polynomial-time operations. Micciancio and Voulgaris [130] presented an alternative deterministic algorithm, which solves both CVP and SVP within $2^{2m+O(m)}$ polynomial-time operations. Interestingly, there are several heuristic variants [72], [140], [129], [192] of AKS with running time $2^{O(m)}$, where $O(.)$

constant, is much less than that of the best provable algorithms known, resulting in the List Sieve algorithms of Micciancio and Voulgaris [129]. Currently, the fastest provable variant of lattice sieving runs in time $2^{2.465m+O(m)}$ and space $2^{1.325m+O(m)}$, [145], (see [96] for a quantum acceleration).
In practice, heuristic variants of the lattice sieving algorithms are found to be more efficient. Nguyen and Vidick [140] exhibited a version of AKS that can be heuristically argued correct and which requires a running-time of $(4/3)^{m+O(m)} \approx 2^{0.4150m+O(m)}$ and space of $(4/3)^{m/2+O(m)} \approx 2^{0.2075m+O(m)}$. Micciancio and Voulgaris [129] later proposed a heuristic variant of their ListSieve algorithm, namely the GaussSieve algorithm. In practice, the GaussSieve seems to perform well compared to the other variants [129]. It has been investigated further in a series of works (see, e.g., [50]). The GaussSieve algorithm is one of the most promising candidates for lattice sieving algorithms in practice.

Nearest neighbor search techniques have been used to accelerate heuristic sieving algorithms further. The technique was first used in the context of lattice sieving by Laarhoven in [95]. Currently, the best variant is due to Becker, Ducas, Gama, and Laarhoven [19], which has a time complexity of$(3/2)^{m/2+O(m)} \approx 2^{0.2925m+O(m)}$ and space complexity $(4/3)^{m/2+O(m)}$.

Recently, Bai, Laarhoven and Stehlé, [16], propose tuple variants for the ListSieve and GaussSieve algorithms which they call TupleSieve and TupleMinkowskiSieve, whose memory footprint is smaller than $2^{0.2075m+O(m)}$. The main idea is to attempt to create shorter vectors by looking at triples, quadruples, etc. of vectors rather than pairs of vectors. For triples of vectors, they estimate the space complexity by $2^{0.1887m+O(m)}$ and for the quadruples, the space complexity is about $2^{0.1724m+O(m)}$.

**Approximation algorithms** These algorithms are much faster than exact algorithms, but they output short lattice vectors, not necessarily the shortest ones: they typically produce a whole reduced basis, and are therefore lattice reduction algorithms. The first algorithm of this kind is the celebrated algorithm of Lenstra, Lenstra and Lovasz (**LLL**) [103], which can approximate SVP to within a factor of $O((2/\sqrt{3})^m)$ in polynomial time. The efficiencies of the three strategies were compared in [140], [199], and simulation results in [140], [129], suggest that for lattices of dimension less than 40, the sphere decoding algorithm using the SE (**Schnorr-Euchner enumeration**) [159] is the most efficient algorithm.

We give a table that summarizes the best known fully analyzed algorithms. We denote by "T", (respectively. "S" and "P or D"), the "Time upper bound" (respectively, the "Space upper bound", "Probabilistic or Deterministic

algorithm").

| | T | S | P or D |
|---|---|---|---|
| via enumeration [70, 71] | $m^{m/2+O(m)}$ | poly$(m)$ | Deterministic |
| via Sieving [145] | $2^{2.247m+O(m)}$ | $2^{1.325m+O(m)}$ | Probabilistic |
| via Voronoi Cell [130] | $2^{2m+O(m)}$ | $2^{m+O(m)}$ | Deterministic |
| via Gaussians [129] | $2^{m+O(m)}$ | $2^{m+O(m)}$ | Probabilistic |

### 2.3.2 The Sphere Decoding Algorithms

#### Idea Behind The Sphere Decoder

The basic premise in Sphere decoding is rather simple: we attempt to search over only lattice points that lie in a certain sphere of radius $\rho$ around the given vector $x$, here in our case $x = 0$, thereby reducing the search space and hence the required computations. Clearly, the shortest lattice vector inside the sphere will also be the shortest lattice vector for the whole lattice. However, close scrutiny of this lattice basic idea leads to two key questions:

**How to choose $\rho$?** Clearly, if $\rho$ is too large, we obtain too many points and the search remains exponential in size, whereas if $\rho$ is too small, we obtain no points inside the sphere.

We have already seen that Minkowski's theorem (see Section. 2.2.7) gives us a simple way to bound the length of the shortest non zero vector. However, $\sqrt{m}\ \det(\mathcal{L})^{\frac{1}{m}}$ can be too large and produces an exponential-time complexity. A natural candidate for $\rho$ is the covering radius of the lattice, defined to be the smallest radius of spheres centered at the lattice points that cover the entire space. This is clearly the smallest radius that guarantees the existence of a point inside the sphere for any vector $x$. The problem with this choice of $\rho$ is that determining the covering radius for a given lattice cannot be solved in polynomial time, [35]. Another choice is to use $\rho$ as the distance between the Babai estimate (found by the Babai nearest plane algorithm [15]. This Babai estimate can be easily obtained by first, finding the real solution for the triangular system $Rz = x$, which is the real least squares solution for the problem $\min \|x - Bz\|_2^2$, then round the entries of $z$ to their nearest integers to obtain the lattice point, $z = \lceil z \rfloor \in \mathbb{Z}^m$) and the vector $x$, i.e., $\rho = \|x - Bz\|$. However, Zhao and Qiao [197] have showed that this method may produce a too small radius and cause sphere decoding to fail to find a solution. Note also that in the case of searching the shortest lattice vector, this distance will be zero.

**How can we know which lattice points are inside the sphere?** If this means testing the distance of each lattice point from $x$ (to determine whether it is less than $\rho$), then there is no point in sphere decoding as we will still need an exhaustive search. A natural candidate for $\rho$ is $\|b_1\|^2$. However, it does propose an efficient way to answer the second. The basic observation is the following: Although it is difficult to determine the lattice points inside a general $m$-dimensional sphere, it is trivial to do so in the (one-dimensional) case of $m = 1$. The reason is that a one-dimensional sphere is simply an interval and so the desired lattice points will be the integer values that lie in this interval. We can use this observation to go from dimension $k$ to dimension $k + 1$. Suppose we have determined all $k$-dimensional lattice points that lie in a sphere of radius $\rho$. Then for any such $k$-dimensional point, the set of admissible values of the $k + 1$th-dimensional coordinate that lie in the higher dimensional sphere of the same radius $\rho$ forms an interval. This means that we can determine all lattice points in a sphere of dimension $m$ and radius $\rho$ by successively determining all lattice points in spheres of lower dimensions $1, 2, ..., m$ and the same radius $\rho$ (see, below).

### 2.3.3 The Algorithms for SVP

Let $\rho$ be the radius of the initial search sphere in which at least one shortest lattice point must lie. The sphere of radius $\rho$ and centered at 0 in (2.7) can be defined as

$$\mathcal{S} = \left\{ z \mid \|\mathbf{B}z\|_2^2 < \rho^2 \right\}, \tag{2.8}$$

Such hypersphere search strategy was firstly presented in [142] and further improved in [5], [159], [49], [191]. To learn the strategy of the hypersphere enumeration, we shall present a recursive version of the sphere decoding algorithms in this section.

Let $\mathbf{R} = \text{chol}(\mathbf{B}^\top \mathbf{B})$ be the Cholesky factorization of $\mathbf{B}^\top \mathbf{B}$, then (2.8) can be transformed into

$$\|\mathbf{R}z\|_2^2 < \rho^2, \tag{2.9}$$

Since $\mathbf{R}$ is an upper triangular matrix, we can rewrite the condition (2.9) as

$$\sum_{i=1}^{m} \left( \sum_{j=i}^{m} r_{i,j} z_j \right)^2 < \rho^2. \tag{2.10}$$

where $r_{i,j}, j \geq i$, denotes the $(i, j)$th entry of $\mathbf{R}$. The above inequality can then be expanded to

$$(r_{m,m} z_m)^2 + (r_{m-1,m-1} z_{m-1} + r_{m-1,m} z_m)^2 + ..... < \rho^2, \tag{2.11}$$

the first entry of $\mathbf{R}z$ is a function of $z_m$ only. We can see that a necessary condition for $\mathbf{R}z$ lying in the hypersphere of radius $\rho$ is $(r_{m,m}z_m)^2 \leq \rho^2$ which is equivalent to the following condition for entry $z_m$:

$$l_m = \left\lceil -\frac{\rho}{r_{m,m}} \right\rceil \leq z_m \leq \left\lfloor \frac{\rho}{r_{m,m}} \right\rfloor = u_m.$$

Partition $\mathbf{R}$ into

$$\begin{bmatrix} \mathbf{R}_{m-1} & h \\ 0^\top & r_{m,m} \end{bmatrix}$$

where $\mathbf{R}_{m-1} \in \mathbb{R}^{(m-1)\times(m-1)}$, $h \in \mathbb{R}^{m-1}$. Then for each integer value of $z_m$, the $m$-dimensional SVP (2.7) is reduced to an $(m-1)$-dimensional CVP.

$$\min \left\{ \|\mathbf{R}_{m-1}z' + z_m h\|_2 : z' \in \mathbb{Z}^{m-1} \right\}, \tag{2.12}$$

with a solution lying in the $(m-1)$-dimensional hypersphere

$$\|\mathbf{R}_{m-1}z' + z_m h\|_2^2 \leq \rho^2 - z_m^2 r_{m,m}^2, \tag{2.13}$$

Therefore, an $m$-dimensional SVP can be reduced to a finite number (at most $\left\lfloor \frac{2\rho}{r_{m,m}} \right\rfloor + 1$) of $(m-1)$-dimensional CVPs, leading to a recursive algorithm. In summary, we unite Pohst's strategy [142], [49], [191] and Schnorr-Euchnerr strategy [5], [159] in the same framework, and present a recursive implementation (see below, details about the recursion implementation). Clearly, a shortest non zero lattice vector can be found by calling Algorithm **Sph-Dec** $(\mathbf{R}, \mathbf{0}, \emptyset, r, 0)$.

**Details About the Recursive Implementation.**

Let

$$c_m = 0,\ c_k = \left( \sum_{j=k+1}^{m} r_{k,j} z_j \right) / r_{k,k},\ \text{for } k \text{ such that} \quad m-1 \leq k \leq 1. \tag{2.14}$$

Note that $c_k$ depends on $z_{k+1}, ..., z_m$. Substituting (2.14) in (2.10), we have

$$\sum_{k=1}^{m} r_{k,k}^2 (z_k - c_k)^2 < \rho^2. \tag{2.15}$$

If $z$ satisfies the bound, then it must also satisfy inequalities

$$\text{level k} : r_{k,k}^2 (z_k - c_k)^2 < \rho^2 - \sum_{i=k+1}^{m} r_{i,i}^2 (z_i - c_i)^2. \tag{2.16}$$

The search process starts at level $m$ and moves down to level 1. At level $k$, $z_k$ is determined for $m-1 \leq k \leq 1$. From (2.16), the range of $z_k$ is $[\ell_k, u_k]$, where

$$\ell_k = \left\lceil c_k - \left( \rho^2 - \sum_{i=k+1}^{m} r_{i,i}^2 \left( z_i - c_i \right)^2 \right)^{\frac{1}{2}} / r_{k,k} \right\rceil$$

and

$$u_k = \left\lfloor c_k + \left( \rho^2 - \sum_{i=k+1}^{m} r_{i,i}^2 \left( z_i - c_i \right)^2 \right)^{\frac{1}{2}} / r_{k,k} \right\rfloor .$$

There are two typical strategies to examine the integers inside $[\ell_k, u_k]$. In the Pohst strategy, the integers are chosen in the ascending order

$$\ell_k, \ell_k + 1, \ell_k + 2, ...., u_k.$$

However, in the Schnorr-Euchner strategy, the integers are chosen in the zig-zag order

$$z_k = \begin{cases} \lfloor c_k \rceil, \lfloor c_k \rceil - 1, \lfloor c_k \rceil + 1, \lfloor c_k \rceil - 2, ..., & \text{if} \quad c_k \leq \lfloor c_k \rceil \\ \lfloor c_k \rceil, \lfloor c_k \rceil + 1, \lfloor c_k \rceil - 1, \lfloor c_k \rceil + 2, ..., & \text{if} \quad c_k \geq \lfloor c_k \rceil . \end{cases}$$

In fact, Schnorr-Euchner strategy combines the advantages of the Babai nearest plane algorithm and the Pohst strategy (when the radius $\rho$ is immediately updated every time a lattice point inside the sphere is found).

Observe that in Schnorr-Euchner strategy, once an integer $z_k$ does not satisfy (2.16), all the following integers in the sequence will not satisfy it. These integers can be pruned from the search process. Such a property does not exist in the Pohst strategy. Another benefit with the Schnorr-Euchner enumeration order is that the first points examined are more likely to minimize (2.16) than the last points examined. As will be seen in the next paragraph, this allows to shrink the search hypersphere faster. Simulations in [5] confirm that the Schnorr-Euchner strategy is more efficient than the Pohst strategy.

We now describe the search process using the Schnorr-Euchner strategy. At level $m$, we set $z_m = 0$. If (2.16) at level $k = m$ is not satisfied, no integer can satisfy (2.15). Otherwise, we go to level $m-1$, compute $c_{m-1}$ and set $z_{m-1} = \lfloor c_{m-1} \rceil$. If (2.16) does not hold, we go back to level $m$ and choose $z_m$ to be the second nearest integer to $c_m$. Otherwise, we move down to level $m-2$. When we reach level 1, we compute $c_1$ and set $z_1 = \lfloor c_1 \rceil$. Then if (2.16) at level $k = 1$ is satisfied, we set $\hat{z} = [z_1, ..., z_m]^\top$, where $\hat{z}$ is a full integer point inside the search hypersphere. We update $\rho$ by setting $l = \rho^2 = \sum_{k=1}^{m} r_{k,k}^2 \left( \hat{z}_k - c_k \right)^2 = nd$. This step allows to eliminate more points

by "shrinking" the search hypersphere. Now we search for a better point than $\hat{z}$. If one is found, we update $\hat{z}$. We move up to level 2 and choose $z_2$ to be the next nearest integer to $c_2$, where "next" is relative to $\hat{z}_2$. If inequality (2.16) holds at level 2, we move down to level 1 and update $z_1$; otherwise, we move up to level 3 and update $z_3$. The procedure continues until we reach level $m$ and (2.16) at level $m$ is not satisfied. The last full integer point found is the OILS (Ordinary Integer Least Squares) or the shortest lattice vector solution.

Note that the efficiency of the sphere decoding algorithm is closely related to the structure of the lattice basis.

Thus, an appropriate preprocessor, such as the LLL algorithm, is useful. The LLL algorithm [103] can be used to reduce the computational complexity in two ways:

First, it can be used to reduce the radius of the search sphere by reducing the norm of $\mathbf{R}$. Second, the sphere decoding is a depth-first searching (i.e., an algorithm that constructs a tree to obtain a solution) for the lattice points inside a sphere, the LLL algorithm can be used to reduce the total number of search paths. Thus, the performance of the sphere decoding algorithm can be further improved for LLL reduced bases.

**Algorithm 1** Sphere-Dec $(R, x, z_{in}, r, \text{dist})$

**Input**: $\mathbf{R} \in \mathbb{R}^{m \times m}$, a vector $x = [x_i] \in \mathbb{R}^m$ to decode, an integer partial solution $z_{in}$, the current distance record $r$ and the distance to the examined layer dist.

**Output**: a solution $\mathbf{z} \in \mathbb{Z}^m$ and $\ell = \|\mathbf{Rz} - x\|_2^2$.

$\mathbf{LB} \leftarrow \left\lceil \frac{-\sqrt{r-\text{dist}}+x_m}{r_{m,m}} \right\rceil$, UB $\leftarrow \left\lfloor \frac{\sqrt{r-\text{dist}}+x_m}{r_{m,m}} \right\rfloor$;

$\ell \leftarrow r$, $\mathbf{z} \leftarrow \emptyset$;

**if** LB $\leq$ UB **then**

  **for** each integer s lying in $[\mathbf{LB}, \mathbf{UB}]$ **do**

    newdist $\leftarrow dist + (x_m - s.r_{m,m})^2$;

    **if** newdist $\leq \ell$ **then**

      $\hat{\mathbf{z}}_{in} \leftarrow [s; \mathbf{z}_{in}]$;

      **if** $m > 1$ **then**

        $\hat{x} \leftarrow x(1 : m-1) - s \times \mathbf{R}(1 : m-1, m)$;

        $[z', \ell'] \leftarrow$ **Sph-Dec** $(\mathbf{R}_{m-1}, \hat{x}, \hat{z}_{in}, \ell, \text{newdist})$;

        **if** $\ell' \leq \ell$ **then**

          $\ell \leftarrow \ell'$, $z \leftarrow z'$;

        **end if**

      **else**

        **if** newdist $\neq 0$ **then**

          $z \leftarrow \hat{z}_{in}$, $\ell \leftarrow$ newdist;

        **end if**

      **end if**

    **end if**

  **end for**

**end if**

Obviously, a shortest non zero vector can be found by calling Algorithm Sph-Dec$(R, 0, \emptyset, r, 0)$ see Algorithm 1. We notice, one $m$-dimensional problem can be solved recursively by reducing it to at most $\lfloor 2\sqrt{r}/r_{m,m} \rfloor + 1$ $(m-1)$-dimensional sub-problems. The size $\ell$ of the search region is reduced dynamically (however, the algorithms based on the Kannan strategy scan all the $(m-1)$-dimensional sub-lattices with the same value of $\rho$). That is, when any lattice point $\mathbf{R}\tilde{z}$ inside the search region is found, the squared radius $\ell$ can be reduced to $\|\mathbf{R}\tilde{z}\|_2^2$, since $\|\mathbf{R}\tilde{z}\|_2^2 < \ell$. Therefore, not all the $\lfloor 2\sqrt{r}/r_{m,m} \rfloor + 1$ $(m-1)$-dimensional sub-problems are necessarily to be solved in practice.

Note that the last condition "if newdist $\neq 0$" is to make sure that the lattice points being searched are non zero. The above algorithm can be applied to solve general CVP, by deleting this last condition.

The complexity of the sphere decoding algorithms was discussed in [73], [80], [140].

The efficiency of the three strategies were compared in [140]. In general, for lattices of small dimensions, the sphere decoding using Schnorr-Euchner enumeration is the fastest. Simulation results in [140], [129] illustrate that for lattices of dimension $m \leq 40$, Schnorr-Euchner enumeration provides the most efficient algorithm.

## 2.4 Lattice reduction:

The history of reduction theory begins in the Euclidean era about 300 BC when the Euclidean algorithm was introduced to compute the greatest common divisor of two integers. A lattice basis reduction algorithms is analogous to Euclid's GCD algorithm. A formal study of integer lattices was initiated in the 18th century by Lagrange, Gauss, Hermite and others. However, little progress was made. As computers became more powerful towards the middle of the 20th century, researchers contributed extensive work to lattices. The development of lattice basis reduction began in 1773 by Lagrange, [40]. He had presented the first algorithm for constructing a reduced bases for lattices of dimension two. The algorithm takes a two-dimensional basis matrix $B$ as input and successively performs column swaps and size reduction until a Gauss reduced basis is obtained where $\|b_1\| \leq \|b_2\|$ and the properties of a size reduced basis are satisfied. Then this algorithm was extended to dimensions three by Vallée in 1986 and Semaev in 2001 [184] [164]: Semaev's algorithm is quadratic without fast integer arithmetic, whereas Vallée's algorithm has cubic complexity. An extension to dimension four by Nguyen and Stehlé [137] was named "greedy algorithm". This implies that a shortest vector and a HKZ

reduced basis can be computed in quadratic time up to dimension four. More generally, Helfrich [74] and Afflerbach and Grothe [3] presented algorithms for lattices of arbitrary dimension. A variant of Kannan's strategy was proposed in [74]. Hence, like Kannan's algorithm [87], [86], Helfrich's algorithm is also intended rather as a theoretical result than as a practical tool and its runtime is exponential, but polynomial for fixed dimension. However, Afflerbach and Grothe handled this problem in a different way: let $p$ be the $p$th stage of the reduction process, starting from $p = 1$, this algorithm first performs Pohst enumeration [49], [142] and during the search process, whenever an intermediate lattice point $B_p z$ inside the search process satisfying the gcd conditions ($\gcd(z_p, ..z_m) = 1$), for $m > 7$ or $z_p = 1$ for $m \leq 7$, is found, the $p$th column of $B_p$ is then replaced by $B_p z$ and the algorithm is restarted from $p = 1$. On the other hand, if the $p$th column of $B_p$ is already the shortest lattice point satisfying the corresponding gcd constraint, we set $p = p + 1$ and repeat the above process. The algorithm terminates when $p = m + 1$. Note that the number of lattice points enumerated by Pohst's strategy grows exponentially with the dimension $m$. Therefore, in practice the algorithm in [3] is restarted many times and the complexity becomes prohibitive quickly as the dimension increases.

**Gauss Reduction.** The reduction method introduced by Gauss in the context of binary quadratic forms is restricted to lattices of rank $m = 2$, [61], which is a natural generalization of the centered Euclidean algorithm, [182]. For such two-dimensional lattices, Gauss reduction constructs a basis that fulfills the reduction criteria introduced by Minkowski and HKZ.

In addition to size reduction, Gauss reduction also includes column swapping operations. In particular, after first size reducing the given basis matrix $B$, the columns of the resulting basis $\tilde{B} = \left(\tilde{b}_1, \tilde{b}_2\right)$ are swapped if the length of $\tilde{b}_1$ is larger than that of $\tilde{b}_2$ and the resulting basis is again size-reduced. This process of successive size reduction and column swapping operations is repeated until the length of $\tilde{b}_1$ is shorter, after the preceding size reduction step, than that of $\tilde{b}_2$, which implies that no further column swapping operation is performed.

After a finite number of iterations, this algorithm provides a **Gauss** reduced basis $\tilde{B}$, where $\left\|\tilde{b}_1\right\| \leq \left\|\tilde{b}_2\right\|$ and the properties of a size reduced basis are satisfied. In particular, $\tilde{b}_1$ and $\tilde{b}_2$ are the two shortest vectors in the lattice $\mathcal{L}$ that form a basis for $\mathcal{L}$.

**Minkowski Reduction.**

**Definition 7.** A lattice generator matrix $B := [b_1, ..., b_m]$ is called Minkowski reduced if for all $1 \leq i \leq m$, the vector $b_i$ has the minimum norm among all lattice vectors $b_i$ such that $\{b_1, ..., b_i\}$ can be extended to a basis for $\mathcal{L}(B)$, [121].

From [187], [94], the length of each Minkowski reduced basis vector can be bounded by

$$\lambda_i^2(\mathcal{L}) \leq \|b_i\|_2^2 \leq \max\left\{1, \left(\frac{5}{4}\right)^{(m-4)}\right\} \lambda_i^2(\mathcal{L}), \ 1 \leq i \leq m; \tag{2.17}$$

$$\prod_{i=1}^{m} \|b_i\|_2 \leq \gamma_m^{\frac{m}{2}} \operatorname{vol}(\mathcal{L}), \text{ for } m \leq 4; \tag{2.18}$$

$$\prod_{i=1}^{m} \|b_i\|_2 \leq \gamma_m^{\frac{m}{2}} \left(\frac{5}{4}\right)^{\frac{(m-3)(m-4)}{4}} \operatorname{vol}(\mathcal{L}), \text{ for } m > 4. \tag{2.19}$$

From (2.17), we deduce that for lattices of dimension $m \leq 4$, the norms of Minkowski reduced basis vectors reach strongly Minkowski's successive minima. However, in high dimensions, there does not necessarily exist a Minkowski reduced basis whose vector norms achieve Minkowski's successive minima. Furthermore, from (2.19) and the Hermite constant, we find that the orthogonality defect of a Minkowski reduced basis is bounded by a constant depending only on the rank of the given lattice.

$$\delta_{M,d} \leq \gamma^{\frac{m}{2}} \left(\frac{5}{4}\right)^{\frac{(m-3)(m-4)}{4}} = \left(\frac{5}{4}\right)^{\frac{m^2}{4}+O(m\log m)}.$$

**Lemma 1.** A lattice generator matrix $B := [b_1, ..., b_m]$ is called a Minkowski reduced basis if and only if for all $1 \leq i \leq m$, and all integers $x_i, ..., x_m$ such that $\gcd(x_i, .., x_m) = 1$, we have:

$$\|x_1 b_1 + ..... + x_m b_m\| \geq \|b_i\|;$$

With the above statement, one might think that to ensure that a given basis is Minkowski reduced, there are infinitely many conditions to be checked. Fortunately, it is sufficient to check a finite subset of them. This result is noted as the second finiteness theorem in [167]. Several sufficient sets of conditions are possible. We call a such a subset with minimal cardinality, *Minkowski conditions*.

For dimensions $m \leq 6$, Minkowski stated a finite number of conditions that a **quadratic form** is Minkowski reduced. In the $m \leq 4$ case, he published a proof in [122]. The proofs in the $m = 5$ and $m = 6$ cases can be found in [2], [152], [153], [176], [177]. Corresponding conditions for the case $m = 7$ were stated and proved in [178]. But for $m \geq 8$ no similar conditions are known to date. Therefore, in low dimension, one can check quickly for $m \leq 7$ if a basis is reduced in the sense of Minkowski by checking these conditions.

Naturally the dimension is limited by $m = 7$. However, in this case already over 90000 conditions have to be checked. Therefore, there is a need for an algorithm which calculates Minkowski reduced bases in acceptable computation time for even greater dimension.

Note that for each $b_i$, there are more than one shortest vector available ($-b_i$ for example). So when we refer to a minimal basis, we mean one among many available minimal bases. As a special case, when $\mathcal{L}$ has a basis consisting of orthogonal vectors, it is automatically a **reduced basis**. In the general case, a reduced basis is the closest to an orthogonal basis that a lattice can have.

**Theorem.** (Minkowski conditions [176], [178])

Let $m \leq 6$. A lattice generator matrix $B := [b_1, ...., b_m]$ is a Minkowski reduced basis if and only if for all $1 \leq i \leq m$ and for all integers $x_i, ....., x_m$ satisfying the three conditions below, we have:

$$\|x_1 b_1 + ..... + x_m b_m\| \geq \|b_i\| ;$$

1. The integers $x_i, ..., x_m$ are relatively prime;

2. For some permutation $\sigma \in \mathbb{S}_m$, $\left(\left|x_{\sigma(1)}\right|, ...., \left|x_{\sigma(m)}\right|\right)$ appears in the list below (where empty places can be counted as zeros);

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 1 | 1 | | | | | |
| 3 | 1 | 1 | 1 | | | | |
| 4 | 1 | 1 | 1 | 1 | | | |
| 5 | 1 | 1 | 1 | 1 | 1 | | |
| | 1 | 1 | 1 | 1 | 2 | | |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 1 | 1 | 1 | 1 | 1 | 2 | |
| | 1 | 1 | 1 | 1 | 2 | 2 | |
| | 1 | 1 | 1 | 1 | 2 | 3 | |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| | 1 | 1 | 1 | 1 | 1 | 2 | 3 |
| | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| | 1 | 1 | 1 | 1 | 2 | 2 | 3 |
| | 1 | 1 | 1 | 1 | 2 | 2 | 4 |
| | 1 | 1 | 1 | 1 | 2 | 3 | 3 |
| | 1 | 1 | 1 | 1 | 2 | 3 | 4 |
| | 1 | 1 | 1 | 2 | 2 | 2 | 3 |
| | 1 | 1 | 1 | 2 | 2 | 3 | 4 |

Moreover, this list is minimal, which means that if a table row is rejected, then a basis satisfy the remaining conditions without being a Minkowski reduced basis.

We have already seen that Minkowski proved a result saying that we only need to check a finite number of inequalities. This is most conveniently expressed in terms of the Gram matrix associated to the basis. Call $B$ the $m \times m$ matrix having $b_i's$ as columns, then the Gram matrix $\mathbf{Q} = B^\top B$ has entries $q_{ij} = q_{ji} = \langle b_i, b_j \rangle$. $\mathbf{Q}$ is a positive definite matrix.

**Theorem** (Minkowski, see [167, 66])**.** Given $m$ linearly independent vectors $B = b_1, ..., b_m$ in $\mathbb{R}^m$. Let $\mathcal{L}$ be the lattice generated by $B$ and $\mathbf{Q}$ be the Gram matrix with $q_{ij} = \langle b_i, b_j \rangle$. Then $B$ is a reduced basis for $\mathcal{L}$ if and only if the $q_{ij}$ entries satisfy a set of linear inequalities, which only depend on the dimension $m$.

We call any symmetric matrix $\mathbf{Q}$ satisfying such inequalities a "reduced or Minkowski" reduced form. Reduction in $\mathbb{R}^2$ is particularly simple and was

known to Gauss. In this case,

$$M = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

is Minkowski reduced when

$$a \leq c \text{ and } \ 2\,|b| \leq a,$$

these correspond to the inequalities

$$\|b_1\|^2 \leq \|b_2\|^2 \ \text{ and } \ 2\,|\langle b_1, b_2\rangle| \leq \|b_1\|^2 .$$

A more geometric way to look at the second inequality is

$$\|b_2\| \leq \|b_1 - b_2\| \ \text{ and } \ \ \|b_2\| \leq \|b_1 + b_2\| ,$$

together with $\|b_1\| \leq \|b_2\|$, these are exactly the finite collection of inequalities for $m = 2$.

In dimension 3, a basis is Minkowski reduced if and only if we have the following inequalities:

$$\begin{gathered} \|b_1\| \leq \|b_2\| \leq \|b_3\| \\ \|b_1 + b_2\| \geq \|b_2\| \quad \|b_1 - b_2\| \geq \|b_2\| \\ \|b_1 + b_3\| \geq \|b_3\| \quad \|b_1 - b_3\| \geq \|b_3\| \\ \|b_2 + b_3\| \geq \|b_3\| \quad \|b_2 - b_3\| \geq \|b_3\| \end{gathered}$$

$$\begin{gathered} \|b_1 + b_2 + b_3\| \geq \|b_3\| \quad \|b_1 + b_2 - b_3\| \geq \|b_3\| \\ \|b_1 - b_2 + b_3\| \geq \|b_3\| \quad \|b_1 - b_2 - b_3\| \geq \|b_3\| . \end{gathered}$$

It was shown in [18] that a $(4 \times 4)$ positive definite matrix

$$M = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{pmatrix}$$

is Minkowski reduced when it verifies the following 39 reduction conditions:

- $a_{11} \leq a_{22} \leq a_{33} \leq a_{44}$;

- For each $1 < i < 4$, we must have $x^\top Mx \geq a_{ii}$ for any $x = \{x_1, x_2, x_3, x_4\}$ satisfying $x_i = 1,\ x_j = 0$ if $j > i$, $x_j \in \{0, 1, -1\}$ if $j < i$, and $x_j \neq 0$ for at least one $j < i$.

The 36 inequalities of the second kind consist of 28 inequalities which correspond to the inequalities of 3 dimensions. Those in fact tell us that the four rank 3 sub-lattices generated by $\{b_2, b_3, b_4\}$, $\{b_1, b_3, b_4\}$, $\{b_1, b_2, b_4\}$ and $\{b_1, b_2, b_3\}$ are also Minkowski reduced. The other eight inequalities are added to compare $\|b_4\|$ with $\|\pm b_1 \pm b_2 \pm b_3 + b_4\|$.

A lattice basis of dimension $m$ that reaches successive minima is necessarily Minkowski reduced, but the inverse isn't true. On the other hand, it still reaches the first four minimum [187] :

If $[b_1, ..., b_m]$ is Minkowski reduced, then for all $1 \leq i \leq \min(m, 4)$, we have $\|b_i\| = \lambda_i(\mathcal{L})$. Thus, a Minkowski reduced basis is optimal in a very natural way up to dimension 4.

**LLL Reduction.** In 1982, a central tool in the algorithmic study of lattices (and their applications) appears, the LLL algorithm of Lenstra, Lenstra and Lovász, which generalizes an old algorithm due to Gauss for reducing lattices of rank 2 to dimensions $m \geq 2$, performed a sequence of steps, each being a translation step, or a swap step. It is a polynomial time algorithm that finds a non zero vector in an $m$-dimensional lattice that is guaranteed to be at most $O((2/\sqrt{3})^m)$-times the length of the shortest non zero vector in that lattice.

This notion of reduction, although not perfect, has solved many problems. After this publication, many results appeared. For example, Kannan provided an algorithm (not polynomial time) for finding the shortest vector of a lattice, from a reduced basis in the sense of Lovász. Babai extends this algorithm to the search of the closest lattice point to a given arbitrary point.

The performance of the LLL algorithm has been further improved by suitable modifications [47], [99], [147], and new algorithms were invented [109], [157], [156], [165], [25], [26], [31].

Seysen [165], [195] and Schnorr [157], [156] have written new algorithms for basis reduction in the square norm. Seysen's method performs extremely well for lattices of dimension up to 30. It operates on small integers. Brun [25], [26], [31], proposed an efficient algorithm for finding approximate integer solutions, i.e., finding integer vectors $t_l \in \mathbb{Z}^m$ that are (almost) orthogonal to a given vector $u \in \mathbb{R}^m$ while being as short as possible. It has been realized in [164] that Brun's algorithm can also be used for lattice reduction. Compared to LLL and Seysen reduction, Brun reduction performs more poorly but has significantly lower complexity.

Schnorr offered many improvements of the LLL algorithm: on the theoretical side, he shows how to make it faster by replacing rationals by floating points in LLL and how to do better by defining a hierarchy of polynomial time algorithms that approximate the shortest vector of a lattice better than the LLL algorithm does but increasingly slow. Finally, together with Euchner [195], he shows how to reduce in practice a lattice as fast as possible. At the same time, the possible applications of the reduced lattice basis algorithms are diversified.

Nguyen and Stehlé, [136], introduced a new and natural floating point variant of the LLL algorithm which provably outputs LLL reduced bases in polynomial time and claim that this is the first LLL algorithm whose running time (without fast integer arithmetic) provably grows only quadratically with respect to $\log C$, ($C$ is a constant where the Euclidean norm of the vectors of the lattice basis are less than $C$), like Euclid's gcd algorithm and Lagrange's two-dimensional algorithm. Also the Jacobi lattice reduction algorithm [146], [181] which presents a new strategy with respect to the LLL algorithm to construct a reduced basis in polynomial time.

Recently, Neumaier and Stehlé, [135], describe an asymptotically fast variant of LLL lattice reduction algorithm. It takes as input a basis $B \in \mathbb{Z}^{m\times m}$ and returns a reduced basis $C$ of the Euclidean lattice $\mathcal{L}$ spanned by $B$, whose first vector satisfies $\|c_1\| \leq (1+c)(4/3)^{\frac{m-1}{4}}(\det \mathcal{L})^{\frac{1}{m}}$ for any fixed $c > 0$. It terminates within $O(m^{4+\epsilon}\beta^{1+\epsilon})$ bit operations for any $\epsilon > 0$, with $\beta = \log \max_i \|b_i\|$.

**Definition.** Let $\delta$ be a constant with $\frac{1}{4} < \delta \leq 1$. A lattice basis $B = \{b_1, ..., b_m\}$ is called LLL reduced if and only if:

$$|\mu_{ij}| \leq \frac{1}{2}, \tag{2.20}$$

$$\delta \left\|b_{i-1}^*\right\|^2 \leq \left\|b_i^* + \mu_{i,i-1}b_{i-1}^*\right\|^2, \text{ for all } i \geq 2. \tag{2.21}$$

It can also be defined as follows:
Let $\mathbf{R}$ be the upper triangular matrix of the $\mathbf{QR}$ decomposition of the basis matrix $B$, then $B$ is LLL reduced if and only if:

$$|r_{ij}| \leq \frac{1}{2}|r_{ii}|, \text{ for } 1 \leq i < j \leq m,$$

$$\delta |r_{j-1,j-1}|^2 \leq |r_{jj}|^2 + |r_{j-1,j}|^2, \ j = 2, ..., m.$$

The choice of the parameter $\delta$ affects the quality of the reduced basis and the computational complexity. We often set $\delta = \frac{3}{4}$.

The second property (2.21) can be written as:

$$\delta \left\| b_{i-1}^* \right\|^2 \leq \left\| b_i^* + \mu_{i,i-1} b_{i-1}^* \right\|^2 = \left\| b_i^* \right\|^2 + \mu_{i,i-1}^2 \left\| b_{i-1}^* \right\|^2 .$$

where the second equality follows since $b_{i-1}^*$ and $b_i^*$ are orthogonal. It follows that

$$\left\| b_i^* \right\|^2 \geq \left( \delta - \mu_{i,i-1}^2 \right) \left\| b_{i-1}^* \right\|^2 \geq \left( \delta - \frac{1}{4} \right) \left\| b_{i-1}^* \right\|^2 .$$

Put this way, the second property (2.21) reads "$b_i^*$ is not much shorter than $b_{i-1}^*$".

Consider the orthogonal basis obtained by normalization of the Gram-Schmidt vectors $b_1^*, ..., b_m^*$. In this basis, $B$ can be written as

$$\begin{bmatrix} \left\| b_1^* \right\| & * & \cdots & * \\ 0 & \left\| b_2^* \right\| & \cdots & * \\ \vdots & & \ddots & \\ & & & \\ & & & \\ 0 & \cdots & & \left\| b_m^* \right\| \end{bmatrix}$$

where column $i$ shows the coordinates of $b_i$ in this orthonormal basis. The first condition (2.20) guarantees that the absolute value of any off-diagonal element is at most half the one in the diagonal element on the same row. This can be written as

$$\begin{bmatrix} \left\| b_1^* \right\| & \leq \frac{1}{2} \left\| b_1^* \right\| & \cdots & \leq \frac{1}{2} \left\| b_1^* \right\| \\ 0 & \left\| b_2^* \right\| & \cdots & \leq \frac{1}{2} \left\| b_2^* \right\| \\ \vdots & & \ddots & \\ & & & \\ & & & \leq \frac{1}{2} \left\| b_{m-1}^* \right\| \\ 0 & \cdots & & \left\| b_m^* \right\| \end{bmatrix}$$

where $\leq \frac{1}{2} \left\| b_j^* \right\|$ indicates that the absolute value of this coordinate is at most $\frac{1}{2} \left\| b_j^* \right\|$. For the second condition (2.21), consider the $2 \times 2$ sub-matrix of the above matrix, with the upper left entry indexed at $(i-1, i-1)$.

$$\begin{bmatrix} \left\| b_{i-1}^* \right\| & \mu_{i,i-1} \left\| b_{i-1}^* \right\| \\ 0 & \left\| b_i^* \right\| \end{bmatrix}.$$

Then the second condition (2.21) requires that the second column of this matrix is almost as long as its first column.

**Proposition.** Let $\{b_1, ..., b_m\} \in \mathbb{R}^n$ be a LLL reduced basis. Then

$$\|b_1\| \leq \left(\frac{4}{4\delta - 1}\right)^{(m-1)/2} \lambda_1(\mathcal{L}).$$

For $\delta = \frac{3}{4}$ this gives

$$\|b_1\| \leq 2^{(m-1)/2} \lambda_1(\mathcal{L}).$$

*Proof.* Since for any basis $b_1, ..., b_m$, $\lambda_1(\mathcal{L}) \geq \min_i \|b_i^*\|$ ($\|b_i\| \geq \|b_i^*\|$). We get

$$\|b_m^*\|^2 \geq (\delta - \frac{1}{4}) \left\|b_{m-1}^*\right\|^2 \geq ..... \geq (\delta - \frac{1}{4})^{m-1} \|b_1^*\|^2 = (\delta - \frac{1}{4})^{m-1} \|b_1\|^2$$

then, for any $i$,

$$\|b_1^*\| \leq (\delta - \frac{1}{4})^{-(i-1)/2} \|b_i^*\| \leq (\delta - \frac{1}{4})^{-(m-1)/2} \|b_i^*\|.$$

Hence,

$$\|b_1\| \leq (\delta - \frac{1}{4})^{-(m-1)/2} \min_i \|b_i^*\| \leq (\delta - \frac{1}{4})^{-(m-1)/2} . \lambda_1(\mathcal{L}).$$

□

**Lemma.** If the LLL procedure described above ever terminates, then its output is an LLL reduced basis for the lattice spanned by the input basis $b_1, ..., b_m$.

*Proof.* We need to prove that the output of the LLL algorithm is a basis for $\mathcal{L}(B)$ that satisfies both properties of an LLL reduced basis. The second property of an LLL reduced basis is enforced by the check during the swap step. The reason that the output of the algorithm is indeed a basis for $\mathcal{L}(B)$, is that we only perform column operations of the form $b_i \leftarrow b_i + ab_j$ for $i \neq j$, with $a \in \mathbb{Z}$. We notice that throughout this step, the Gram-Schmidt basis does not change. After this step of reduction, $b_1, ..., b_m$ satisfy $|\mu_{i,j}| \leq \frac{1}{2}$, for all $i > j$. Consider some $i > j$, then $|\mu_{i,j}|$ can be written as

$$|\mu_{i,j}| = \left|\frac{\left\langle b_i - c_{i,j}.b_j, b_j^* \right\rangle}{\left\langle b_j^*, b_j^* \right\rangle}\right| = \left|\frac{\left\langle b_i, b_j^* \right\rangle}{\left\langle b_j^*, b_j^* \right\rangle} - \left\lceil \frac{\left\langle b_i, b_j^* \right\rangle}{\left\langle b_j^*, b_j^* \right\rangle} \right\rfloor . \frac{\left\langle b_j, b_j^* \right\rangle}{\left\langle b_j^*, b_j^* \right\rangle}\right| \leq \frac{1}{2}.$$

Where the first equality follows from the fact that $\mu_{i,j}$ was $|\mu_{i,j}| \geq \frac{1}{2}$ at the beginning $\left(c_{i,j} = \left\lceil \frac{\left\langle b_i, b_j^* \right\rangle}{\left\langle b_j^*, b_j^* \right\rangle} \right\rfloor\right)$, and the last inequality follows from the fact that $\left\langle b_j, b_j^* \right\rangle = \left\langle b_j^*, b_j^* \right\rangle$. □

**Proposition.** Let $b_1, ..., b_m$ be an LLL reduced basis for a lattice $\mathcal{L}$ in $\mathbb{R}^n$, $\delta = 3/4$ and let $b_1^*, ..., b_m^*$ be the Gram-Schmidt orthogonalization. Then we have

$$\|b_j\|^2 \leq 2^{i-1} \|b_i^*\| \text{ for } i \leq j \leq i \leq m, \tag{2.22}$$

$$\det(\mathcal{L}) \leq \prod_{i=1}^{m} \|b_i\| \leq 2^{m(m-1)/4} \det(\mathcal{L}), \tag{2.23}$$

$$\|b_1\| \leq 2^{(m-1)/4} \det(\mathcal{L})^{\frac{1}{m}}. \tag{2.24}$$

*Proof.* From the LLL conditions, we notice that

$$\|b_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \left\|b_{i-1}^*\right\|^2 \geq \frac{1}{2} \left\|b_{i-1}^*\right\|^2,$$

for $1 < i \leq m$, so by induction we get

$$\left\|b_j^*\right\|^2 \leq 2^{i-j} \|b_i^*\|^2 \text{ for } 1 \leq j \leq i \leq m.$$

Now, From the Gram-Schmidt orthogonalization and condition (2.20) we obtain

$$\begin{aligned}
\|b_i\|^2 &= \|b_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{ij}^2 \left\|b_j^*\right\|^2 \\
&\leq \|b_i^*\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} 2^{i-j} \|b_i^*\|^2 \\
&= \left(1 + \frac{1}{4}\left(2^i - 2\right)\right) \|b_i^*\|^2 \\
&\leq 2^{i-1} \|b_i^*\|^2.
\end{aligned}$$

It follows that

$$\|b_j\|^2 \leq 2^{j-1} \left\|b_j^*\right\|^2 \leq 2^{i-1} \|b_i^*\|^2$$

for $1 \leq j \leq i \leq m$. This proves (2.22).
The determinant $\det(\mathcal{L})$ of $\mathcal{L}$ is defined by

$$\det(\mathcal{L}) = |\det(b_1, ..., b_m)|,$$

It follows from the Gram-Schmidt orthogonalization that

$$\det(\mathcal{L}) = |\det(b_1^*, ..., b_m^*)|$$

and thus, since the $b_i^*$ are pairwise orthogonal

$$\det(\mathcal{L}) = \prod_{i=1}^{m} \|b_i^*\|.$$

Note that the left side of the second inequality follows from "Hadamard's inequality" [67], [78]. Now, from $\|b_i^*\| \leq \|b_i\|$ and $\|b_i\| \leq 2^{(i-1)/2} \|b_i^*\|$, we deduce the second property (2.23).
Putting $j = 1$ in (2.22) and taking the product over $i = 1, 2, ..., m$ we find (2.24). □

**Proposition 1.** Let $\mathcal{L}$ be a lattice with reduced basis $b_1, ..., b_m$. Then

$$\|b_1\|^2 \leq 2^{m-1} \|x\|^2,$$

for every $x \in \mathcal{L}$, $x \neq 0$.

*Proof.* $x \in \mathcal{L}$ , then $x = \sum_{i=1}^{m} r_i b_i = \sum_{i=1}^{m} r_i' b_i^*$ with $r_i \in \mathbb{Z}$, $r_i' \in \mathbb{R}$, $(1 \leq i \leq m)$. And as, $x \neq 0$ so if we take $i$ as the largest index with $r_i \neq 0$ then $r_i' = r_i$, so

$$\|x\|^2 \geq r_i'^2 \|b_i^*\|^2 \geq \|b_i^*\|^2.$$

By using (2.22) and putting $j = 1$, we obtain

$$\|b_1\|^2 \leq 2^{i-1} \|b_i^*\|^2 \leq 2^{m-1} \|b_i^*\|^2.$$

This proves the proposition. □

Note that, by proposition 1 we can also prove a previous result, $\|b_1\| \leq 2^{(m-1)/2} \lambda_1(\mathcal{L})$.

**Proposition 2.** Let $\mathcal{L}$ be a lattice with reduced basis $b_1, ..., b_m$. And let $x_1, x_2, ...., x_t \in \mathcal{L}$ be linearly independent. Then we have

$$\|b_j\|^2 \leq 2^{m-1} \max\left\{\|x_1\|^2, \|x_2\|^2, ..., \|x_t\|^2\right\}$$

for $j = 1, 2, ..., t$.

*Proof.* We have $x_j = \sum_{i=1}^{m} r_{ij} b_i$ with $r_{ij} \in \mathbb{Z}$ $(1 \leq i \leq m)$ for $1 \leq j \leq t$. For fixed $j$, let $i(j)$ be the largest $i$ for which $r_{ij} \neq 0$. Then, as in the previous proof of proposition 1, we have

$$\|x_j\|^2 \geq \left\|b_{i(j)}^*\right\|^2, \tag{2.25}$$

for $1 \leq j \leq t$. We renumber the $x_j$ such that $i(1) \leq i(2) \leq ... \leq i(t)$ and we claim that $j \leq i(j)$ for $1 \leq j \leq t$. If not, we take for example $j = 2$ and $j > i(j)$, hence $i(2) = 1$, then we notice that $r_{i,2} = 0$ for $2 \leq i \leq m$ which implies that $x_2$ belongs to $\mathbb{R}b_1$, so we conclude that in this case $x_1, ...., x_t$ would all belong to $\mathbb{R}b_1 + ... + \mathbb{R}b_{j-1}$, a contradiction with the linear independence of $x_1, ...., x_t$. Therefore, from $j \leq i(j)$, (2.22) and using (2.25) we obtain

$$\begin{aligned} \|b_j\|^2 &\leq 2^{i(j)-1} \left\|b^*_{i(j)}\right\|^2 \\ &\leq 2^{m-1} \left\|b^*_{i(j)}\right\|^2 \\ &\leq 2^{m-1} \|x_j\|^2, \end{aligned}$$

for $j = 1, 2, ..., t$. This proves the proposition. □

Let now, $\lambda_1, \lambda_2, ..., \lambda_m$ denote the successive minima on $\mathcal{L}$ and $b_1, ..., b_m$ a reduced basis for $\mathcal{L}$. Then by the previous proposition 2 and (2.22), we deduce that

$$2^{1-i}\lambda_i^2 \leq \|b_i\|^2 \leq 2^{m-1}\lambda_i^2, \text{ for } 1 \leq i \leq m.$$

So, the norm $\|b_i\|^2$ is a reasonable approximation of $\lambda_i^2$.

**Definition** (Complex LLL Reduction)**.** Let $H = \{h_1, ..., h_m\}$ an $n \times m$ complex lattice basis and $\{h_1^*, ..., h_m^*\}$ the set of orthogonal vectors generated by the Gram-Schmidt orthogonalization. $H$ is CLLL reduced (complex LLL-reduced), if both of the following conditions are satisfied:

$$|\mathcal{R}(\mu_{ij})| \leq \frac{1}{2} \text{ and } |\mathcal{I}(\mu_{ij})| \leq \frac{1}{2},$$

for $1 \leq j < i \leq m$, and

$$H_k \geq \left(\delta - |\mu_{k,k-1}|^2\right) H_{k-1},$$

where $H_k$ denote the squared norm of $h_k^*$, i.e, $H_k = \|h_k^*\|^2$, for $1 < k \leq m$, and $\delta$ with $\frac{1}{2} < \delta < 1$ is a factor selected to achieve a good quality-complexity tradeoff, see [103], [58], for more details.

The LLL algorithm has many applications. Here is a brief description of some of these applications.

1. Approximation to the SVP and the CVP (see, [15, 5]).

2. Finding $\mathbb{Z}$-linear relations among real numbers (Machin's formula) [172].

3. cryptanalysis: breaking cryptosystems based on number theory (breaking the Merkle-Hellman cryptosystem...).

4. The simple application of LLL to algorithmic number theory is the two-square theorem: if $p$ is a prime $\equiv 1 \bmod 4$, the $p$ is a sum of two squares $p = x^2 + y^2$.

5. Integer programming, factoring polynomials (see, [104]) and many more...

**HKZ Reduction** In reduction theory, we often distinguish between reductions that are weak, but can be computed efficiently, and reductions that are strong but that require a much larger amount of computational resources. The famous reduction of the first family is the LLL reduction (can be reached in polynomial-time). However, the famous one in the second family is the HKZ reduction.

Note that an HKZ reduced basis is LLL reduced for any $1/4 < \delta < 1$. There are two main algorithms to compute an HKZ reduced basis. The first one is due to Kannan [84] and further refined by Helfrich and Schnorr [74, 157]. Its complexity has been revisited by Hanrot and Stehlé [70] who proved a $m^{\frac{m}{2e}(1+O(1))}$ upper bound, where $m$ is the lattice dimension. The other algorithm is due to Ajtai, Kumar and Sivakumar (AKS) [9]. It was introduced as the first single-exponential time algorithm for shortest lattice vector problem, However, no explicit time bound was given. In [148], Regev described a simple version of this algorithm, running in time $2^{16m+O(m)}$. The constant in the exponent was decreased from 16 to 5.9 by Nguyen and Vidick [140], 3.4 by Micciancio and Voulgaris [129], 2.7 by Pujol and Stehlé [145] and the currently best time complexity upper bound is $2^{2.465m+O(m)}$ with a space requirement bounded by $2^{1.325m+O(m)}$ [72]. The latter algorithm has a much better asymptotic complexity upper bound than Kannan's. All algorithms based on the Kannan strategy are intended as theoretical results and the complexity quickly becomes prohibitive as the dimension of the lattice increases. However, the second algorithm had also drawbacks (which is a Monte-Carlo probabilistic algorithm running in exponential time and space).

Recently, Zhang, Qiao and Wei, [198], have proposed a new algorithm for constructing a HKZ basis. They used Schnorr's strategy and a different method for the extension of a shortest vector into a new lattice basis (the unimodular transformation technique presented in [112]) than the basis extension strategy introduced by Kannan (which only works for rational lattices, not for general real lattices).

**Definition 8.** A lattice basis $B = \{b_1, .., b_m\}$ is called HKZ reduced if the upper triangular matrix $R$ of the $QR$ decomposition of $B$ is size reduced and for each trailing $(m-i+1)$-by-$(m-i+1)$ sub-matrix, $1 \leq i < m$, its first column is a shortest non zero vector in the lattice generated by the sub-matrix.

**HKZ Properties:**

An HKZ reduced basis has a very interesting property, it provides a very good approximation of the successive minima of a lattice.

**Theorem 4.** If $[b_1, .., b_m]$ is an HKZ basis of a lattice $\mathcal{L}$, then

$$\frac{4}{i+3}\lambda_i^2(\mathcal{L}) \leq \|b_i\|^2 \leq \frac{i+3}{4}\lambda_i^2(\mathcal{L}) \text{ for } 1 \leq i \leq m,$$

Note that the upper bound of this theorem is due to K. Mahler, [119].

*Proof.* Consider $\mathcal{L}^{m-i+1} = \pi_i(\mathcal{L})$, the lattice of rank $m-i+1$. From the definition of successive minima $\lambda_i(\mathcal{L})$ and under the projection $\mathcal{L} \longrightarrow \mathcal{L}^{(m-i+1)}$, at least one of them maps to a non zero vector. Therefore we have

$$\lambda_1(\mathcal{L}^{m-i+1}) = \|b_i^*\| \leq \lambda_i(\mathcal{L}).$$

Now, from the Gram-Schmidt orthogonalization and the size reduction, we obtain the right side of the inequality

$$\|b_i\|^2 \leq \|b_i^*\|^2 + \frac{1}{4}\sum_{j=1}^{i-1}\left\|b_j^*\right\|^2 \leq \lambda_i^2(\mathcal{L}) + \frac{1}{4}\sum_{j=1}^{i-1}\lambda_j(\mathcal{L}) \leq \frac{i+3}{2}\lambda_i^2(\mathcal{L}).$$

For the left side of the inequality, since $\pi_j(b_i)$ is a non zero element of $\mathcal{L}^{(m-j+1)}$ and for $j \leq i$ we have

$$\left\|b_j^*\right\|^2 = \lambda_1^2(\mathcal{L}^{(n-j+1)}) \leq \|\pi_j(b_i)\|^2 \leq \|b_i\|^2.$$

The last inequality comes from the fact that $b_i = b_i^* + \sum_{k=1}^{i-1}\mu_{ki}b_k^*$ and $\pi_j(b_i) = b_i^* + \sum_{k=j}^{i-1}\mu_{ki}b_k^*$.
Hence, for $j \leq i$ we have

$$\|b_j\|^2 \leq \left\|b_j^*\right\|^2 + \frac{1}{4}\sum_{k=1}^{j-1}\|b_k^*\|^2 \leq \frac{j+3}{4}\|b_i\|^2.$$

Therefore, we obtain

$$\lambda_i^2(\mathcal{L}) \leq \max\left\{\|b_j\|^2 : 1 \leq j \leq i\right\} \leq \frac{i+3}{4}\|b_i\|^2 .$$

□

**Theorem.** If $[b_1, ..., b_m]$ is a HKZ basis of a lattice $\mathcal{L}$, then

$$\prod_{i=1}^{m}\|b_i\|^2 \leq \left(\gamma_m^m \prod_{i=1}^{m}\frac{i+3}{4}\right)\det(\mathcal{L})^2.$$

*Proof.* This follows from the previous theorem. 4 and Minkowski's theorem, (see 2.2.7):

$$\prod_{i=1}^{m}\lambda_i(\mathcal{L}) \leq \gamma_m^{\frac{m}{2}}.\det(\mathcal{L}).$$

□

**Proposition 3.** Let $[b_1, .., b_m]$ be a HKZ basis of a lattice $\mathcal{L}$, and let $\mathcal{L}^\star$ be its reciprocal (dual) lattice. Then we have

$$\|b_i\|^2 \lambda_1(\mathcal{L}^\star) \leq \frac{i+3}{4}\gamma_m^{\star 2}$$

for $1 \leq i \leq m$, where $\gamma_m^\star$ is defined by $\max\{\gamma_i\}$, $\gamma_i$ is Hermite's constant for a lattice of dimension $i$ (see, [98]).

*Proof.* Since $\mathcal{L}^{(m-j+1)\star}$ is a sub-lattice of $\mathcal{L}^\star$, we have $\lambda_1(\mathcal{L}^\star) \leq \lambda_1(\mathcal{L}^{(m-j+1)\star})$ for each $j$. From the Gram-Schmidt decomposition and HKZ definition, we have

$$\|b_i\|^2 \leq \|b_i^*\|^2 + \frac{1}{4}\sum_{j=1}^{i-1}\left\|b_j^*\right\|^2 = \lambda_1(\mathcal{L}^{(m-i+1)})^2 + \frac{1}{4}\sum_{j=1}^{i-1}\lambda_1(\mathcal{L}^{(m-j+1)})^2.$$

Therefore, we obtain

$$\|b_i\|^2 \lambda_1(\mathcal{L}^\star)^2 \leq \lambda_1(\mathcal{L}^{(m-i+1)})^2\lambda_1(\mathcal{L}^{(m-i+1)\star})^2 + \frac{1}{4}\sum_{j=1}^{i-1}\lambda_1(\mathcal{L}^{(m-j+1)})^2\lambda_1(\mathcal{L}^{(m-j+1)\star})^2.$$

By the definition of Hermite's constant (see, section. Minkowski's successive minimum and Hermite's constant), we have that for any lattice $N$ of rank $k$

$$\lambda_1(N)^2\lambda_1(N^\star)^2 \leq \gamma_k.\det(N)^{\frac{2}{k}}.\gamma_k.\det(N^\star)^{\frac{2}{k}} = \gamma_k^2,$$

We deduce that

$$\|b_i\|^2 \lambda_1(\mathcal{L}^\star)^2 \leq \gamma_{m-i+1}^2 + \frac{1}{4}\sum_{j=1}^{i-1} \gamma_{n-j+1}^2 \leq \frac{i+3}{4}\gamma_m^{\star 2}.$$

□

**Proposition 4.** For any lattice $\mathcal{L}$ of rank $m$ with reciprocal lattice $\mathcal{L}^\star$ we have

$$\lambda_i(\mathcal{L})^2 \lambda_1(\mathcal{L}^\star)^2 \leq \frac{i+3}{4}\gamma_m^{\star 2}$$

for $1 \leq i \leq m$.

*Proof.* This follows from the previous proposition. 3, since

$$\lambda_i(\mathcal{L})^2 \leq \max\left\{\|b_j\|^2 : 1 \leq j \leq i\right\}.$$

□

**Theorem 5.** If $[b_1, ..., b_m]$ is HKZ basis of a lattice $\mathcal{L}$, then

$$\|b_i\|^2 \lambda_{m-i+1}(\mathcal{L}^\star)^2 \leq \frac{i+3}{4}\frac{n-i+4}{4}\gamma_m^{\star 2}, \text{ for } 1 \leq i \leq m.$$

*Proof.* Since $\mathcal{L}^{(m-j+1)\star}$ is a sub-lattice of $\mathcal{L}^\star$, we have

$$\lambda_{m-i+1}(\mathcal{L}^\star) \leq \lambda_{m-i+1}(\mathcal{L}^{(m-j+1)\star})$$

whenever $j \leq i$. Combining this with

$$\|b_i\|^2 \leq \|b_i^*\|^2 + \frac{1}{4}\sum_{j=1}^{i-1}\left\|b_j^*\right\|^2 = \lambda_1(\mathcal{L}^{(m-i+1)})^2 + \frac{1}{4}\sum_{j=1}^{i-1}\lambda_1(\mathcal{L}^{(m-j+1)})^2,$$

we obtain,

$$\begin{aligned}\|b_i\|^2 \lambda_{m-i+1}(\mathcal{L}^\star)^2 \leq{}& \lambda_1(\mathcal{L}^{(n-i+1)})^2 \lambda_{m-i+1}(\mathcal{L}^{(m-i+1)\star})^2 \\ &+ \frac{1}{4}\sum_{j=1}^{i-1}\lambda_1(\mathcal{L}^{(n-j+1)})^2 \lambda_{m-i+1}(\mathcal{L}^{(m-j+1)\star})^2.\end{aligned}$$

Applying the previous proposition. 4 to each $\mathcal{L}^{(m-j+1)}$ we find that

$$\begin{aligned}\|b_i\|^2 \lambda_{m-i+1}(\mathcal{L}^\star)^2 \leq{}& \frac{m-i+4}{4}\gamma_{m-i+1}^{\star 2} + \frac{1}{4}\sum_{j=1}^{i-1}\frac{m-i+4}{4}\gamma_{m-j+1}^{\star 2} \\ &\frac{m-i+4}{4}\frac{i+3}{4}\gamma_m^{\star 2}.\end{aligned}$$

□

**Theorem.** The successive minima of a lattice $\mathcal{L}$ of rank $m$ and its dual lattice $\mathcal{L}^\star$ satisfy

$$1 \leq \lambda_i(\mathcal{L})^2 \lambda_{m-i+1}(\mathcal{L}^\star)^2 \leq \frac{i+3}{4}\frac{m-i+4}{4}\gamma_m^{\star 2},$$

for $1 \leq i \leq m$.

*Proof.* The lower bound is due to [28]. For the upper bound, interchanging $\mathcal{L}$ and $\mathcal{L}^\star$, if necessary, we may assume that $i \leq (m+1)/2$. Choosing an HKZ basis $[b_1, .., b_m]$ of $\mathcal{L}$ and applying the previous theorem. 5, we obtain

$$\begin{aligned}
\lambda_i(\mathcal{L})^2 \lambda_{m-i+1}(\mathcal{L}^\star) &\leq \max\left\{\|b_i\|^2 : 1 \leq j \leq i\right\} \lambda_{m-i+1}(\mathcal{L}^\star)^2 \\
&\leq \max\left\{\|b_j\|^2 \lambda_{m-j+1}(\mathcal{L}^\star)^2 : 1 \leq j \leq i\right\} \\
&\leq \max\left\{\frac{j+3}{4}.\frac{m-j+4}{4}\gamma_m^{\star 2} : 1 \leq j \leq i\right\} \\
&= \frac{i+3}{4}\frac{m-i+4}{4}\gamma_m^{\star 2}.
\end{aligned}$$

□

## 2.5 An introduction to the fundamental domain of Minkowski reduction

Suppose that $f = \sum_{i,j=1}^m a_{ij} x_i x_j$ is a positive definite quadratic form with real coefficients $a_{ij}$. The condition for $f$ to be Minkowski-reduced is that, for all $i = 1, .., m$, and for all integer $(l_1, .., l_m)$ if

$$\gcd(l_i, .., l_m) = 1 \text{ then } f(l_1, ..., l_m) \geq a_{ii}.$$

Hermite reduction is closely related to Minkowski reduction. A form $f$ is Hermite reduced if for $\delta \to \infty$ the function

$$h(f, \delta) = a_{11}\delta^{m-1} + a_{22}\delta^{m-2} + .... + a_{mm}$$

is the smallest among functions defined in the same way for all forms equivalent to the form $f$, [137]. We notice from the definition that corresponding diagonal coefficients of two equivalent Hermite reduced forms are identical.
The domains $\mathcal{H}$ for Hermite's reduction and $\mathcal{M}$ for Minkowski reduction are convex. $\mathcal{H} = \mathcal{M}$ for $m \leq 6$, [154] [177] and different for $m > 6$, [150] (a part of the boundary of $\mathcal{M}$ doesn't belong to $\mathcal{H}$).

The positivity cone $\wp$ corresponds to the set of positive-definite quadratic forms in the $N = \frac{1}{2}m\,(m+1)$-dimensional space of the coefficients ($a_{11}$,.., $a_{mm}$, $a_{12}$,.., $a_{m-1,m}$).

The set of all Minkowski reduced forms in $\wp$ will be called the classical Minkowski region, and its subset with the additional constraint

$$a_{i,i+1} \geq 0 \ (i = 1, .., m)$$

will be called the simple Minkowski reduction region.

The Minkowski reduction regions (classical and simple) are convex gonohedra with finitely many planar faces in the positivity cone $\wp$ (finitely many inequalities of Minkowski's condition suffice to define it), [123]. These finitely many inequalities have been determined for $m \leq 7$, [151, 150, 176, 178, 123].

Consider the set of regions equivalent to any of the reduction regions of quadratic forms. The set of such regions covers the entire $\wp$ and different regions have no common interior points. Only the classical Minkowski region may have completely coinciding equivalent regions; other regions are fundamental and may not coincide. The set of equivalent regions forms a partition of the positivity cone.

We call a partition of the positivity cone $\wp$ normal if any $N$-dimensional element of the partition touches only one element of the partition along any integral $(N-1)$-dimensional face (it suffices to show that the Minkowski reduction region has at least one $(N-1)$-dimensional face which is partitioned into two $(N-1)$-dimensional parts, such that an integral unimodular transformation moves one part of this face into itself and removes the other part outside the Minkowski reduction region). Otherwise, the partition is called non normal.

Note that a face of dimension $(N-k)$ for $1 \leq k \leq N-1$ is any non empty set $\Gamma$ of points of the Minkowski domain of reduction $\mathcal{M}$ which satisfies the following properties:

1. $\Gamma$ lies in some $(N-k)$-dimensional plane.
2. $\Gamma$ is open in this plane.
3. $\Gamma$ contains no points of the faces of dimensions $N, ..., N-k+1$.
4. $\Gamma$ is maximal: if $\Gamma \subset \Gamma' \subset \mathcal{M}$ and $\Gamma'$ satisfies conditions 1, 2, 3, then $\Gamma' = \Gamma$.

The unique face of dimension $N$ is the interior of the cell $\mathcal{M}$.

A simple domain of Minkowski reduction together with its equivalent domains fills out the cone of positivity without gaps and superpositions of interior points. It gives a partition of the cone of positivity.

For $m = 2$, the simple domain of Lagrange-Minkowski reduction is

$$\begin{cases} a_{12} \geq 0 \\ a_{11} - 2a_{12} \geq 0 \\ a_{22} \geq a_{11} > 0 \end{cases}$$

Tammela showed in [176] that for $m \geq 3$ the partition of the cone of positivity into regions (domains) equivalent to a simple Minkowski reduction region (to a simple domain of Minkowski reduction) is non normal (some domains intersect along pieces of the face), the same for $m \geq 7$ in [179], Tammela showed that for $m \geq 7$ the partition of the cone of positivity into regions equivalent to the classical Minkowski reduction region is non normal.

**Theorem** ([176])**.** Let $m \geq 3$. Then the partition of the cone of positivity into domains equivalent to a simple domain of Minkowski reduction is not normal.

*Proof.* We will give here the main steps of the proof without going into details. Let $\mathcal{D}$ be the simple Minkowski reduction region. In order to prove the theorem, it suffices to find an $(N-1)$-dimensional closed face $\Gamma$ of $\mathcal{D}$, two quadratic forms $f_1$ and $f_2$, and an integral unimodular matrix $S$ which satisfy the following conditions: 1) $f_1$ lies inside $\Gamma$, 2) $f_2 \in \Gamma$, 3) $f_1 S = g_1$ lies inside $\Gamma$, 4) $f_2 S = g_2 \notin \mathcal{D}$. Then the region $\mathcal{D}$ and $\mathcal{D}S$ touch at the interior points of the face $\Gamma$, but $\mathcal{D} \cap \mathcal{D}S \neq \Gamma$. Note that $(fS)(x) = f(Sx)$, $\mathcal{D}S = \{f' = fS \mid f \in \mathcal{D}\}$. □

### 2.5.1 Equivalence of reduced quadratic forms

We consider a form $f$ from the domain of Hermite reduction.

**Definition.** We denote by $\mathcal{Z}_1(f)$ the collection of all integer vectors

$$\begin{pmatrix} l_1^{(i_1)} \\ \cdot \\ \cdot \\ l_m^{(i_1)} \end{pmatrix},$$

for which

$$f(l_1^{(i_1)}, ..., l_m^{(i_1)}) = a_{11}.$$

Let $k$ be an index with $2 \leq k \leq m$, we denote by $\mathcal{Z}_k(f)$ the collection of all integral primitive matrices (the matrix is primitive if the greatest common divisor of the minors of order $k$ of this matrix is equal to one)

$$\begin{pmatrix} l_1^{(i_1)} & . & . & l_1^{(i_{k-1})} & l_1^{(i_k)} \\ . & . & . & . & . \\ l_m^{(i_1)} & . & . & l_m^{(i_{k-1})} & l_m^{(i_k)} \end{pmatrix},$$

where

$$\begin{pmatrix} l_1^{(i_1)} & . & . & l_1^{(i_{k-1})} \\ . & . & . & . \\ l_m^{(i_1)} & . & . & l_m^{(i_{k-1})} \end{pmatrix} \in \mathcal{Z}_{k-1}(f)$$

and

$$f(l_1^{(i_k)}, ..., l_m^{(i_k)}) = a_{kk}.$$

We denote $\mathcal{Z}_m(f)$ by $\mathcal{Z}(f)$. By the definition of the domain of Hermite reduction, $\mathcal{Z}(f)$ is the set of all transformations of the form $f$ into forms which are Hermite reduced.

Let $f$ be a Hermite reduced form, and let $S$ be an integral unimodular matrix. Then by the definition of Hermite reduction in order that $fS$ be Hermite reduced it is necessary and sufficient that

$$f(s_{i1}, ..., s_{im}) = a_{ii}, \quad i = (1, ..., m).$$

Therefore, if for some form $f$ and integral, unimodular matrix $S$ for some $k = 1, ..., m$

$$f(s_{i1}, ..., s_{im}) = a_{ii}, \quad i = (1, ..., k-1)$$

and

$$f(s_{k1}, ..., s_{km}) < a_{kk},$$

then the form $f$ is not Hermite reduced.

**Theorem.** Let $\Gamma$ be a face of the domain of Hermite reduction of dimension $d$ $(1 \leq d \leq N)$ and let $f, g \in \Gamma$. Then

$$\mathcal{Z}_k(f) = \mathcal{Z}_k(g) \quad (k = 1, ..., m).$$

We can deduce from this theorem that the set $\mathcal{Z}_k(f)$ only depends on $\Gamma$ and does not depend on the choice of $f \in \Gamma$. Therefore, we denote

$$\mathcal{Z}_k(\Gamma) = \mathcal{Z}_k(f) \quad (k = 1, ..., m), \quad \mathcal{Z}(\Gamma) = \mathcal{Z}(f).$$

By Minkowski's theorem ($|l_{ik}| < C$), we deduce that the set $\mathcal{Z}(\Gamma)$ is finite.

**Theorem.** Let $\Gamma$ be a face of the domain of Hermite reduction of dimension $d$ $(1 \leq d \leq N)$ and let $f \in \Gamma$. In order that an integer, unimodular matrix $S = (s_{ij})$ transforms the form $f$ into a Hermite reduced form it is necessary and sufficient that $S \in \mathcal{Z}(\Gamma)$.

For $m \leq 6$ the construction of $\mathcal{Z}(\Gamma)$ can be found in [176].
Let now $\Gamma$ be a face of a simple domain of Hermite reduction of dimension $d$ $(1 \leq d \leq N)$. We consider a form $f \in \Gamma$.

**Definition.** We denote by $\mathcal{Z}_1^{(0)}(f)$ the collection of all integer vectors

$$\begin{pmatrix} l_1^{(i_1)} \\ . \\ . \\ l_m^{(i_1)} \end{pmatrix},$$

for which

$$f(l_1^{(i_1)}, ..., l_m^{(i_1)}) = a_{11}.$$

If $k$ is an index with $2 \leq k \leq m$, we denote by $\mathcal{Z}_k^{(0)}(f)$ the collection of all integer primitive matrices

$$\begin{pmatrix} l_1^{(i_1)} & . & . & l_1^{(i_{k-1})} & l_1^{(i_k)} \\ . & . & . & . & . \\ l_m^{(i_1)} & . & . & l_m^{(i_{k-1})} & l_m^{(i_k)} \end{pmatrix},$$

where

$$\begin{pmatrix} l_1^{(i_1)} & . & . & l_1^{(i_{k-1})} \\ . & . & . & . \\ l_m^{(i_1)} & . & . & l_m^{(i_{k-1})} \end{pmatrix} \in \mathcal{Z}_{k-1}^{(0)}(f)$$

and

$$f(l_1^{(i_k)}, ..., l_m^{(i_k)}) = a_{kk},$$

and

$$f(l_1^{(i_{k-1})}, .., l_m^{(i_{k-1})}; l_1^{(i_k)}, ..., l_m^{(i_k)}) \geq 0.$$

We denote $\mathcal{Z}_m^{(0)}(f)$ by $\mathcal{Z}^{(0)}(f)$.

$\mathcal{Z}^{(0)}(f)$ is the set of all transformations of the form $f$ into forms $f'$ belonging to a simple domain of Hermite reduction.

Note that if $f$ is considered as a form from the classical domain of Hermite reduction, i.e., if $f \in \Gamma$, where $\Gamma$ is a face of a classical Hermite domain of dimension $d$, then $\mathcal{Z}^{(0)}(f) \subset \mathcal{Z}(\Gamma)$.

**Theorem 6.** Interior points of a simple domain of reduction have only trivial transformations (the automorphisms $\pm I$, where $I$ is the identity transformation) into forms that belong to a simple domain of reduction.

*Proof.* Let $f = (a_{ij}) \in \Gamma_0$ be an interior form of a simple domain of Hermite reduction, and let $f' = (a'_{ij})$ be a form of a simple domain of Hermite reduction equivalent to $f$. As $\mathcal{Z}^{(0)}(f)$ is contained among matrices that transform the interior points of the classical domain of Hermite reduction into points of the classical domain of Hermite reduction, i.e., by the previous theorem among matrices of the form

$$\begin{pmatrix} \epsilon_1 & & 0 \\ & \cdot & \\ & & \cdot \\ 0 & & \epsilon_n \end{pmatrix}$$

where $\epsilon_i = \pm 1 \quad (i = 1, .., m)$. But since $a_{i,i+1} > 0$, (interior points), $a_{i,i+1} \geq 0 \quad (i = 1, ..., m-1)$, it follows that $\epsilon_1 = \epsilon_2 = ... = \epsilon_m = \pm 1$. Therefore $\mathcal{Z}^{(0)}(f) = \{\pm I\}$. □

Let $\Gamma$ be a face of a simple domain of Hermite reduction of dimension $d$ $(1 \leq d \leq N)$ and let $g \in \Gamma$. We consider a partition of $\Gamma$ by the planes

$$f(s_{i1}, ..., s_{im}; s_{i+1,1}, ..., s_{i+1,m}) = 0, \quad (i = 1, ..., m-1),$$

where $S = (s_{ij}) \in \mathcal{Z}^{(0)}(g)$. We obtain partitions

$$\Gamma = \bigcup_{k=1}^{p} \Gamma_k,$$

where $\Gamma_k$ are the parts of $\Gamma$ of different dimensions which are open in the respective subspace.

Let the points $f, g \in \Gamma_k$, then $\mathcal{Z}^{(0)}(f) = \mathcal{Z}^{(0)}(g)$ for otherwise there would exist a plane (defined above) separating these points. Therefore, we can say

$$\mathcal{Z}^{(0)}(\Gamma_k) = \mathcal{Z}^{(0)}(f) \quad \text{if } f \in \Gamma_k.$$

The finiteness of this partition follows from the Minkowski theorem $|s_{ik}| < c$ for transformations $S = (s_{ij})$ of a domain of Minkowski reduction into equivalent domains having common points with a domain of Minkowski reduction not lying on the boundary of the cone of positivity.

Let $L = \{\Gamma_i\}_{i=0}^{M}$ be a finite set that represents the elements of the partition of all faces. Note that points of faces of a classical domain of Hermite reduction can be equivalent only to points of faces of a classical domain of Hermite

reduction of the same dimension, we find that $\Gamma_i$ is either equivalent to $\Gamma_j$ $(j \neq i;\ j = 0, ...., M)$ or has no equivalent points with $\Gamma_j$ $(j \neq i;\ j = 0, ..., M)$.

A simple domain of reduction can be considered as the basis for constructing a fundamental domain $\mathcal{F}$ of reduction of positive quadratic forms (i.e, as a set of forms such that any positive definite form is equivalent to one and only one form of the fundamental domain).

Note that the definition of the faces of a fundamental domain is analogous to the definition of the faces of a simple domain of Hermite reduction.

### 2.5.2 The exact domain of Minkowski reduction for m= 3

Tammela described in [176] an algorithm for finding a fundamental domain. It starts by a partition $L = \{\Gamma_i\}_{i=0}^{M}$ of all the faces of a simple domain of Hermite reduction (where $\Gamma_0$ represent the interior of a simple domain of Hermite reduction). He constructed a set $\mathcal{F}$ as the union $\mathcal{F} = \bigcup_{i=0}^{M} \mathcal{F}_i$ of certain subsets $\mathcal{F}_i = \bigcup_p \Gamma_i^{(p)}$ of a simple domain of Hermite reduction, where the $\Gamma_i^{(p)}$ are parts of the set $\Gamma_i$. For constructing $\mathcal{F}_k$ $(k = 0, ..., M)$, it was enough to consider $\Gamma_k$, where three cases were possible:
a) $\Gamma_k$ is equivalent to one of the $\Gamma_j$, $j < k$; for this $\mathcal{F}_k = \emptyset$; b) $\Gamma_k$ is not equivalent to one of the $\Gamma_j$, $j < k$, and all transformations of $\Gamma_k$ into itself are automorphisms of all the forms of $\Gamma_k$, then $\mathcal{F}_k = \Gamma_k$; c) $\Gamma_k$ is not equivalent to one of the $\Gamma_j$, $j < k$, but among the transformations of $\Gamma_k$ into itself there are transformations $S = (s_{ij})$ which are not automorphisms of all the forms of $\Gamma_k$; this case was considered separately.
Let $S^{(t)} = (S_{ij}^{(t)})$ $(t = 1, ..., \iota)$ be all the transformations in question in part c). The partition of $\Gamma_k$ by the planes

$$f(S_{i1}^{(t)}, ...., S_{im}^{(t)}; S_{j1}^{(t)}, ..., S_{jm}^{(t)}) = a_{ij} \quad (t = 1, .., \iota; i, j = 1, ..., m).$$

is considered into account and provides the following partition

$$\Gamma_k = \bigcup_{p=1}^{v} \Gamma_k^{(p)},$$

where $\Gamma_k^{(p)}$ are parts of $\Gamma_k$ of different dimensions.
If $f \in \Gamma_k^{(p)}$ and suppose that some transformation $S^{(t)}$ $(1 \leq t \leq \iota)$ takes $f$ into the form $f' \in \Gamma_k^{(p)}$, then $S^{(t)}$ is an automorphism of the form $f$, since otherwise there would exist a plane (defined above) separating the equivalent forms $f$ and $f'$. Then Tammela defined

$$\mathcal{F}_k = \bigcup_{p \in P} \Gamma_k^{(p)},$$

where $p$ runs through the set $P \subset [1, ..., v]$ which is constructed as follows: $1 \in P$; $h \in P$ if $\Gamma_k^{(h)}$ is not equivalent to one of the sets $\Gamma_k^{(j)}$, $j < h$; and otherwise $h \notin P$.
Having thus considered all $i = 0, ..., M$, corresponding to elements $\Gamma_i$ of the partition $L$, Tammela constructed the sets $\mathcal{F}_i$ $(i = 0, ..., M)$. Then

$$\mathcal{F} = \bigcup_{i=0}^{M} \mathcal{F}_i.$$

For $m = 2$ the simple domain of Lagrange-Minkowski reduction is also a fundamental domain. (It suffices to prove that all forms of the domain are not equivalent. Since an interior form of the domain is not equivalent to any other reduced form, it suffices to consider the boundary points. All transformations of the form $f$ from the domain into reduced forms are automorphisms of this form, for more details see [176]).

However, for $m = 3$ this is no longer true: on the boundary of a simple Hermite-Minkowski domain there are equivalent points.

**Theorem.** The following 16 systems of linear inequalities between the coefficients of forms define a fundamental domain:

$$\begin{cases} a_{12} > 0 \\ a_{23} > 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \end{cases} \qquad \begin{cases} a_{12} = 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{13} \geq 0 \end{cases}$$

$$\begin{cases} a_{12} = 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} = a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{23} \geq a_{13} \\ a_{13} \geq 0 \end{cases} \qquad \begin{cases} a_{12} > 0 \\ a_{23} = 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{13} \geq 0 \end{cases}$$

$$\begin{cases} a_{12} \geq 0 \\ a_{23} = 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} = a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{12} \geq a_{13} \\ a_{13} \geq 0 \end{cases} \qquad \begin{cases} a_{12} > 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} = 0 \\ a_{11} - 2a_{13} \geq 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} \geq a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ 2a_{23} \geq a_{13} \\ a_{13} \geq 0 \end{cases}$$

$$\begin{cases} a_{12} > 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} = 0 \\ a_{11} - 2a_{13} \geq 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} = a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{13} \geq 2a_{23} \end{cases} \qquad \begin{cases} a_{12} \geq 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} = 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} \geq 0 \\ a_{22} \geq a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ 2a_{23} \geq a_{12} \end{cases}$$

$$\begin{cases} a_{12} \geq 0 \\ a_{23} > 0 \\ a_{11} - 2a_{12} \geq 0 \\ a_{11} - 2a_{13} \geq 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} = 0 \\ a_{22} > a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ 2a_{13} \geq a_{12} \end{cases}$$

$$\begin{cases} a_{12} \geq 0 \\ a_{23} > 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} \geq 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} = 0 \\ a_{22} > a_{11} \\ a_{33} = a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{13} \geq 2a_{12} \end{cases}$$

$$\begin{cases} a_{12} > 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} = a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} \geq 0 \\ a_{23} \geq a_{13} \\ a_{13} \geq -a_{23} \end{cases}$$

$$\begin{cases} a_{12} \geq 0 \\ a_{23} > 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} = a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{12} \geq |a_{13}| \end{cases}$$

$$\begin{cases} a_{12} \geq 0 \\ a_{23} \geq 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} = a_{11} \\ a_{33} = a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} \geq 0 \\ a_{12} \geq |a_{13}| \\ a_{23} \geq a_{12} \end{cases}$$

$$\begin{cases} a_{12} \geq 0 \\ a_{23} > 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} \geq 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} = 0 \\ a_{22} = a_{11} \\ a_{33} = a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} > 0 \\ a_{13} \geq 2a_{12} \end{cases}$$

$$\begin{cases} a_{12} > 0 \\ a_{23} > 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} = a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} = 0 \\ a_{12} \geq -a_{13} \\ a_{11} - a_{12} + 2a_{13} \leq 0 \end{cases} \qquad \begin{cases} a_{12} > 0 \\ a_{23} > 0 \\ a_{11} - 2a_{12} > 0 \\ a_{11} - 2a_{13} > 0 \\ a_{11} + 2a_{13} > 0 \\ a_{22} - 2a_{23} > 0 \\ a_{22} > a_{11} \\ a_{33} > a_{22} \\ a_{11} + a_{22} - 2a_{12} + 2a_{13} - 2a_{23} = 0 \\ a_{11} - a_{12} + 2a_{13} \leq 0 \end{cases}$$

**Theorem.** If $Y = (y_{kl})_{1 \leq k,l \leq m}$ is a Minkowski reduced matrix, then the following holds:

- $y_{k,k} \leq y_{k+1,k+1}$ $(k = 1, ..., m-1)$,
- $|2y_{k,l}| \leq y_{k,k}$ $(k < l)$,
- there exists a real number $c_m$ such that

$$\det(Y) \leq \prod_{v=1}^{m} y_{v,v} \leq c_m . \det(Y)$$

where $c_m$ is a constant depending only on $m$.

A proof can be found on page 13 of Klingen's book, [90]. The last inequality is called "Minkowski's inequality" (see below).

### Minkowski's fundamental inequality

Minkowski established the existence of a number $c_m$ with the property that, if $f(x) = \sum_1^m a_{i,j} x_i x_j$ is positive definite and reduced in the sense of Minkowski, with determinant $D = \det(a_{i,j})$, then

$$a_{11} a_{22} ... a_{mm} \leq c_m D,$$

named Minkowski's fundamental inequality for reduced quadratic forms.

Lekkerkerker (1969, Section 10) [100] and Van der Waerden (1956) [185] give detailed accounts of reduction theory and the best estimates for $c_m$ in this "fundamental inequality". Mahler has made several contributions to the

theory of Minkowski reduction. In particular, he obtained in (1938) [116] an estimate for $c_m$ for all $m$, applicable to general convex bodies, and in (1940) [117] and (1946) [118] he gave proofs of the best possible results for $m = 3$ and $m = 4$. Best possible results are now known for $m \leq 5$; these are

$$c_2 = \frac{4}{3}, c_3 = 2, c_4 = 4, c_5 = 8.$$

So in fact for all $m \leq 5$, $c_m = \gamma_m^m$ where $\gamma_m$ is Hermite's constant; for $m = 5$, see Van der Waerden (1969) [186] and Nelson (1974) [134].

# Chapter 3

# Time Complexity Of Reduction Algorithms

This chapter is intended to study the complexity of each reduction algorithm without going into the details of the computational complexity theory, which is a branch of the theory of computation in theoretical computer science.
We provide some background on complexity theory in sections 3.1, 3.2 and 3.3. We present the algorithms that produce the reduction notions cited in the previous chapter and comment their performance and computational complexity in sections 3.5, 3.6 and 3.7.

## 3.1 Mathematical Preliminaries

We start by explaining mathematical preliminaries that are important for some details in this chapter.

**Turing Machine:** In theoretical computer science, a Turing Machine is a theoretical machine that is used in thought experiments to examine the abilities and limitations of computers.
A Turing Machine uses a tape, which is considered to be infinite in both directions. The tape consists of a series of squares each of which can hold a simple symbol. The tape head or read-write head, can read a symbol from the tape, write a symbol to the tape and move one square in either direction. In other words, A Turing Machine is a 7-tuple $(Q, \Sigma, \Gamma, \Delta, q_0, B, F)$ whose components have the following meaning:

- $Q$: The finite set of states of the machine;
- $\Sigma$: The finite set of input symbols;

- $\Gamma$: The complete set of tape symbols;
- $\Delta$: The transition function. The arguments of $\Delta(q, X)$ are a state $q$ and a tape symbol $X$. The value of $\Delta(q, X)$, if it is defined, is a triple $(p, Y, D)$ where

1. $p$: is the next state in $Q$;
2. $Y$ is the symbol, in $\Gamma$, written in the cell being scanned, replacing whatever symbol was there;
3. $D$ is the direction, either $L$ or $R$, standing for "left" or "right", respectively and telling us the direction in which the head moves;

- $q_0$: The start state, a member of $Q$, in which the machine is found initially;
- $B$: The blank symbol. This symbol is in $\Gamma$ but not in $\Sigma$;
- $F$: The set of final or accepting states; a subset of $Q$.

There are two kinds of Turing Machines: a deterministic and non-deterministic Turing Machine.

**Deterministic Turing Machine,** [163]: A Turing Machine is deterministic if there is only one possible action each step.

**Non-Deterministic Turing Machine,** [163]: A Turing Machine is non-deterministic if there is many finite actions each step.

**Probabilistic Turing Machine** (PTM), [155]: The finite state machine in the Probabilistic Turing Machine is probabilistic, in other words, a transition can be a random choice according to fixed, predetermined probabilities.
For example, the transition function will look something like the following:

$$\Delta(q, a) = \begin{cases} q_1, b, L & \text{with probability} \frac{1}{2} \\ q_2, c, R & \text{with probability} \frac{1}{2} \end{cases}$$

In this case, the transition function will transition from state $q$ to state $q_1$, write $a$ $b$, and move its head left with probability $\frac{1}{2}$ and it will transition to state $q_2$, write $a$ $c$, and move its head right with probability $\frac{1}{2}$.

**Oracle Turing Machine,** [12]: An Oracle is a language $A$. An Oracle Turing Machine, is a standard Turing Machine with an additional tape denoted as the oracle tape. The machine can copy characters onto the oracle tape and in a single step receive definitive knowledge of whether the string is in the language $A$.

**Definition.** Given computational problems $A$ to $B$, $A$ is *reducible* to $B$ if: given a way instantly finding answers to arbitrary instances of $B$ allows for some easy method of solving arbitrary instances of $A$. The source of information concerning $B$ is called an *oracle* to $B$. The algorithm for using the answers of $B$ in order to find answers to $A$ is called the *reduction algorithm.*

## 3.2 Introduction

How long does this sorting program run? By the computational complexity (or for short, complexity) of an algorithm, we mean the number of basic computational steps (such as arithmetical operations and comparisons) required for its execution. This number clearly depends on the size and nature of the input.
Since the algorithm's performance may vary with different types of input data, hence for an algorithm we usually use the worst-case time complexity of an algorithm because that is the maximum time taken for any input size.

**Definition.** The worst-case time complexity of an algorithm is expressed as a function

$$\mathcal{T} : \mathbb{N} \longrightarrow \mathbb{N}$$

where $\mathcal{T}(n)$ is the maximum number of steps in any execution of the algorithm on inputs of size $n$. The amount of time an algorithm takes depends on how large is the input on which the algorithm must operate: sorting large lists takes longer than sorting short lists; multiplying huge matrices takes longer than multiplying small ones.

We have already defined the worst-case time complexity so for example, if the time complexity of an algorithm is $3.n^2$, it means that on inputs of size $n$, the algorithm requires up to $3.n^2$ steps. To make this precise, we must classify what we mean by **input size** and **step**.

**Definition.** We define the **size of the input** in a way that is problem-

dependent. For example, if we are dealing with algorithms for multiplying square matrices, we may express the input size as the dimension of the matrix (i.e, the number of columns, or rows), or we may express the input size as the number of entries in the matrix. Sometimes there may be several reasonable choices for the size of input.
So, in order to properly interpret the function that describes the time complexity of an algorithm we must be clear about how exactly we measure the size of inputs.

**Definition.** In general we will consider a **step** to be anything that we can reasonably expect a computer to do in a fixed amount of time. Typical examples are performing an arithmetic operations, comparing two numbers.

The question that we can ask now, how can we define time complexity in an universal way?
Thus, we measure time in somewhat abstractly defined **steps**, there is little point in fretting over the precise number of steps. If by some definition of steps the time complexity of the algorithm is $5.n^2$, by a different definitions of steps it might be $7.n^2$, and by yet another definition of steps it might be $n^2/2$. For this, computer scientists have developed some special notation about functions, known as the **big-oh**, the **big-omega** and **big-theta** notation.

If $k \in \mathbb{N}$, $\mathbb{N}^{\geq k}$ denotes the set of natural numbers that are greater than or equal to k. $\mathbb{R}^{\geq 0}$ denotes the set of non negative real numbers and $\mathbb{R}^{>0}$ denotes the set of positive real numbers.

**Definition.** Let $f : \mathbb{N}^{\geq k} \longrightarrow \mathbb{R}^{\geq 0}$, for some $k \in \mathbb{N}$. $O(f)$ is the following set of functions from $\mathbb{N}^{\geq l}$ to $\mathbb{R}^{\geq 0}$, for any $l \in \mathbb{N}$: O(f) :=

$$\Big\{\text{g: there exist c} \in \mathbb{R}^{>0} \text{ and } n_0 \in \mathbb{N} \text{ such that for all } n \geq n_0,\ \text{g}(n) \leq c.f(n)\Big\}.$$

Therefore, g $\in$ O(f) if for all sufficiently large $n$ (for $n \geq n_0$), g($n$) is bounded from above by $f(n)$ - possibly multiplied by a positive constant. We say that $f$ is an asymptotic upper bound for g.

**Example.** $f(n) = 3.n^2 + 4.n^{\frac{3}{2}} \in \text{O}(n^2)$. This is because $3.n^2 + 4.n^{\frac{3}{2}} \leq 3.n^2 + 4.n^2 \leq 7.n^2$. Thus, take $n_0 = 0$ and $c = 7$. For all $n \geq n_0$, $f(n) \leq c.n^2$.

If the complexity is bounded from above by a polynomial in the input size, the algorithm is called a polynomial-time algorithm. Such an algorithm

is further qualified as linear-time, quadratic-time and so on.
There is a similar notation for asymptotic lower bounds, the **big-omega** notation.

**Definition.** Let $f : \mathbb{N}^{\geq k} \longrightarrow \mathbb{R}^{\geq 0}$, for some $k \in \mathbb{N}$. $\Omega(f)$ is the following set of functions from $\mathbb{N}^{\geq l}$ to $\mathbb{R}^{\geq 0}$, for any $l \in \mathbb{N}$: $\Omega(f)$:=

$$\Big\{\text{g: there exist } d \in \mathbb{R}^{>0} \text{ and } m_0 \in \mathbb{N} \text{ such that for all } n \geq m_0,\ \text{g}(n) \geq d.f(n)\Big\}$$

Therefore, g $\in \Omega(f)$ if for all sufficiently large $n$ (for $n \geq m_0$) g$(n)$ is bounded from below by $f(n)$-possibly multiplied by a positive constant. We say $f(n)$ is an asymptotic lower bound for g$(n)$.

**Definition.**

$$\Theta(f) := \mathrm{O}(f) \cap \Omega(f).$$

Thus, if $g(n) \in \Theta(f)$ then $g(n)$ and $f(n)$ are within a constant factor of each other.

## 3.3 Computational Complexity

Computational complexity theory is concerned with how much computational resources are required to solve a given task.

**Definition** (Polynomial-time solvable algorithms)**.** The significance of polynomial-time algorithms is that they are usually found to be computationally feasible, even for large input.

By contrast, algorithms whose complexity is exponential in the size of the input have running times which render them unusable even her inputs of moderate size. Researchers recognized early on that not all problems can be solved this quickly, but had a hard time figuring out exactly which ones could and which ones could not. There a several so-called **NP-hard** problems, which most people believe cannot be solved in polynomial time, even though nobody can prove a super-polynomial lower bound.
In this connection, a class of problems denoted by **NP** (standing for non-deterministic polynomial-time) plays an important role. We give here an informal definition of this class: a precise treatment can be found in the book

of Garey and Johnson, [59], or in chapter 29 of the Handbook of Combinatories, [64], [14].

**The classes P, NP and co-NP:**

An alphabet is a finite set of symbols $\Sigma$. A string (over $\Sigma$) is a sequence of symbols from $\Sigma$ (a string of bits is a finite sequence of zeroes and ones). The length of a string $y$ is the number of symbols in y, and it is denoted by $|y|$. The set of all strings over $\Sigma$ is denoted by $\Sigma^*$ and the set of all strings of length $n$ is denoted by $\Sigma^n$.
A Turing Machine M runs in time $\mathcal{T}(n)$, if for every input string $\omega$ of length $n$ (over some fixed input alphabet $\Sigma$), M($n$) stops after at most $\mathcal{T}(n)$ steps. We identify the notion of efficient computations with Turing Machines that stop in time polynomial in the size of the input, i.e, Turing Machines that run in time $\mathcal{T}(n) = a + n^b$ for some constants $a$, $b$ independent of $n$.
There are many basic problems for which polynomial time algorithms have yet to be found, and indeed might well not exist. Determining which problems are solvable in polynomial time and which are not is evidently a fundamental question. For this reason, let us define three classes of decision problems.

**Definition** (Decision Problem)**.** A decision problem is the problem of deciding whether the input string satisfies or not some specified property. Formally, a decision problem is specified by a language, i.e, a set of strings $L \subseteq \Sigma^*$, and the problem is given an input string $\omega \in \Sigma^*$ decide whether $\omega \in L$ or not. Thus, a decision problem is a question whose answer is either yes or no.

**Definition** (P)**.** The class of decision problems that can be solved by a deterministic Turing Machine in polynomial time is called **P**. Such a problem belongs to the class P if there is a polynomial time algorithm that solves any instance of the problem in polynomial time. We can say that P is the set of problems that can be solved quickly.

**Definition** (NP)**.** The class of decision problems that can be solved by a non-deterministic Turing Machine in polynomial time is called **NP**. Such a problem belongs to the class NP if, given any instance of the problem whose answer is yes, there is a certificate validating this fact which can be checked in polynomial time; such a certificate is said to be succinct (that means that you can quickly check whether a candidate solution is a solution). Intuitively, NP is the set of problems where we can verify a Yes answer quickly if we have

the solution in front of us.

**Definition** (co-NP)**.** Analogously, a decision problem belongs to the class **co-NP** if, given any instance of the problem whose answer is no, there is a succinct certificate which confirms that this is so. (A decision problem $X$ is a member of co-NP if and only if its complement $\overline{X}$ is in the complexity class NP. Instances of decision problems in co-NP are sometimes called "counterexamples", or co-NP is the set of decision problems where "no" instances can be solved in polynomial time by a theoretical non-deterministic Turing machine).

We can notice from those definitions that P $\subseteq$ NP. Likewise, P $\subseteq$ co-NP. Therefore, P $\subseteq$ NP $\cap$ co-NP.

**Conjecture.** P $\neq$ NP.

**Conjecture.** P= NP $\cap$ co-NP.

The first one is one of the most fundamental open questions in mathematics. (A prize of one million dollar has been offered for its resolution). It is widely (but not universally) believed that the conjecture is true, that there are problems in NP for which no polynomial time algorithms exists.
The second one is strongly supported by empirical evidence. Most decision problems which are known to belong to NP $\cap$ co-NP are also known to belong to P. A particular case is the problem of deciding whether a given integer is prime. Although it had been known for some time that this problem belongs to both NP and co-NP, a polynomial time algorithm for testing primality was discovered only much more recently by Agrawal, Kayal and Saxena, [4].

**Reducing Problems** Sometimes, a common approach to problem-solving is to transform the given problem into one whose solution is already known, and then convert that solution into a solution of the original problem. Of course, this approach is feasible only if the transformation can be made rapidly.

A reduction from problem X to problem Y means that if we have an algorithm for problem Y, we can use it to find an algorithm for problem X.

**Using Reductions:**

A) we use reductions to find algorithms to solve problems.

B) we also use reductions to show that we cannot find algorithms for some problems. (We say that these problems are "hard").

We say that X is polynomial-time reducible or reducible in polynomial time to Y if the reduction map between them can be computed in polynomial time. Therefore, we say X is polynomially reducible to Y and write $X \leq Y$. The significance of polynomial reducibility is that $X \leq Y$, and if there is a polynomial time algorithm for solving Y, then this algorithm can be converted into polynomial time algorithm for solving X.
If X and Y are both decision problems;

$$X \leq Y \text{ and } X \in P \Longrightarrow Y \in P.$$

**Definition** (Karp reduction)**.** Let A and B be two decision problems. A *Karp reduction* from A to B is a polynomial time computable function $f : \Sigma^* \to \Sigma^*$ such that $x \in$ A if and only if $f(x) \in$ B.

Clearly, if A reduces to B can be solved in polynomial time, then also A can be solved in polynomial time. Let us define another notion of reductions called **Cook reduction** and **Randomized reduction**.

**Definition.** A *Cook Reduction* from A to B is a polynomial time Turing Machine M with access to an oracle that takes instances of problem B as input. M reduces A to B, if, given an oracle that correctly solves problem B, M correctly solves problem A.
A problem A is *NP-Hard under Cook Reduction* if for any NP problem B there is a Cook Reduction from B to A. If A is NP, then we say that A is *NP-Complete under Cook Reduction.*

**Definition.** Language A reduces to language B *under randomized reduction*, denoted $A \leq_r B$, if there exists a deterministic, polynomial time computable function $f : \{0,1\}^* \to \{0,1\}^*$ and a polynomial $p()$ such that

$$\text{For all x} \in \text{A},\ \Pr_{y \in \{0,1\}^{p(|x|)}}[f(x,y) \in B] \geq \frac{2}{3}$$

$$\text{For all x} \notin \text{A},\ \Pr_{y \in \{0,1\}^{p(|x|)}}[f(x,y) \in B] \leq \frac{1}{3}$$

In other words, the reduction from A to B is a polynomial time computable function by a probabilistic algorithm $f : \Sigma^* \to \Sigma^*$ such that $x \in$ A if and only if $f(x) \in$ B. The output of the reduction is only required to be correct with sufficiently high probability, for more details see [128].

**NP-Complete Problems.** NP-complete problems are special problems in class NP, i.e., a subset of class NP. A problem $p$ is NP-complete if:

1. $p \in$ NP (you can solve it in polynomial time by a non-deterministic Turing Machine),

2. All other problems in class NP can be reduced to problem $p$ in polynomial time.

This means that the NP-complete problems are the most difficult problem in class NP. If we could solve just one of them in a polynomial time, we could solve all problems in class NP in a polynomial time.

**NP-Hard Problems** are partly similar but more difficult problems than NP-complete problems. They do not themselves belong to class NP (or they do, but no one has found it yet) but all problems in class NP can be reduced to them, i.e., a problem satisfying the second condition of NP-Complete problems is said to be "NP-Hard", whether or not it satisfies the first condition of NP-Complete problems. Therefore, a **NP-Hard problem** cannot be solved in polynomial time unless P = NP.
We notice that NP-Complete problems are a subset of NP-Hard problems, for more details, see [69]. The standard technique to prove that a problem A in NP-Hard (and therefore no polynomial time solution for A is likely to exists) is to reduce some other NP-Hard problem B to A.

## 3.4 Complexity of the Gauss algorithm

The proof that the algorithm terminates is based on the fact that the value of $\left\|\tilde{b}_2\right\|$ decreases strictly. A lattice is a discrete set, therefore, it can take only a finite number of values. We show now, that this algorithm computes a good basis, realizing the first and the second lattice successive minima. First, the algorithm produces necessarily a basis because at each step, the applied transformation is unimodular. Secondly, as the obtained basis satisfies $\left\|\tilde{b}_1\right\| \leq \left\|\tilde{b}_2\right\|$ and $2\left|\tilde{b}_1.\tilde{b}_2\right| \leq \left\|\tilde{b}_1\right\|^2$. Therefore, for any vector of

**Algorithm 2** Gauss reduction

**Input**: a two dimensional basis matrix $B = (b_1, b_2)$.
**Output**: a **Gauss** reduced basis $\tilde{B} = \left(\tilde{b}_1, \tilde{b}_2\right)$.

$B_1 = \|b_1\|^2$;
$\mu = \langle b_1, b_2 \rangle / B_1$;
$b_2 = b_2 - \lfloor \mu \rceil \, b_1$;
$B_2 = \|b_2\|^2$;
**while** $B_2 < B_1$ **do**
  Swap $b_1$ and $b_2$;
  $\tilde{b}_1 = b_2$ and $\tilde{b}_2 = b_1$;
  $\tilde{B}_1 = \left\|\tilde{b}_1\right\|^2 = B_2$;
  $\mu = \left\langle \tilde{b}_1, \tilde{b}_2 \right\rangle / \tilde{B}_1$;
  $\tilde{b}_2 = \tilde{b}_2 - \lfloor \mu \rceil \, \tilde{b}_1$;
  $\tilde{B}_2 = \left\|\tilde{b}_2\right\|^2$;
**end while**

the form $\alpha\tilde{b}_1 + \beta\tilde{b}_2$, where $\alpha$ and $\beta$ are integers, we have:

$$\begin{aligned}
\left\|\alpha\tilde{b}_1 + \beta\tilde{b}_2\right\|^2 &= \alpha^2 \left\|\tilde{b}_1\right\|^2 + 2\alpha\beta\tilde{b}_1.\tilde{b}_2 + \beta^2 \left\|\tilde{b}_2\right\|^2, \\
&\geq \left(\alpha^2 - |\alpha\beta| + \beta^2\right) \left\|\tilde{b}_1\right\|^2, \\
&\geq \left(\left(|\alpha| - |\beta|\right)^2 + |\alpha\beta|\right) \left\|\tilde{b}_1\right\|^2, \\
&\geq \left\|\tilde{b}_1\right\|^2.
\end{aligned}$$

Thus, $\tilde{b}_1$ is the shortest lattice vector. Now, if $\tilde{b}_2^*$ is the perpendicular projection of $\tilde{b}_2$, we have $\tilde{b}_2^* = \tilde{b}_2 - \mu\tilde{b}_1$, with $|\mu| \leq \frac{1}{2}$, therefore $\frac{3}{4}\left\|\tilde{b}_2\right\|^2 \leq \left\|\tilde{b}_2^*\right\|^2 \leq \left\|\tilde{b}_2\right\|^2$, $\left(\left\|\tilde{b}_2^*\right\|^2 = \left\|\tilde{b}_2\right\|^2 - \mu^2\left\|\tilde{b}_1\right\|^2\right)$.

We will move now to check that $\tilde{b}_2$ realizes the second successive minimum of the lattice and for it we choose $\alpha\tilde{b}_1 + \beta\tilde{b}_2$, a linearly independent vector from $\tilde{b}_1$, i.e, with $\beta \neq 0$. The length of this vector is greater than $\beta^2\left\|\tilde{b}_2^*\right\|^2$ and therefore to $\frac{3\beta^2}{4}\left\|\tilde{b}_2\right\|^2$. As soon as, $|\beta| > 1$, this vector is greater than $\tilde{b}_2$, therefore, even changing the signs of $\alpha$ and $\beta$, it can be assumed that $\beta = 1$, and in this case, the last step of Gauss's algorithm, ensures that the vector $\tilde{b}_2$ is the shortest one among these of the form $\tilde{b}_2 - \mu\tilde{b}_1$. All this proves, that the algorithm calculates a lattice basis, which realizes the successive minima.

The Gauss algorithm performs a number of iterations that is linear in the

size of the entries. For an input basis of size $M$, the number of iterations is at most $\log_{\sqrt{3}} M + 2$.

The proof is not trivial, as we need for it a $t$-Gauss algorithm, where the stop condition is replaced by a stronger one: the $\left\|\tilde{b}_1\right\| \leq \left\|\tilde{b}_2\right\|$ condition is replaced by $\left\|\tilde{b}_1\right\| \leq t \left\|\tilde{b}_2\right\|$ for $t > 1$. The proof has two steps:

First, we show that the $t$-Gauss algorithm terminates in polynomial number of iterations in the size of the entries. Then we show that the Gauss algorithm is at most one more iteration than the $t$-Gauss, for a $t$ well chosen, for details, see [190].

We illustrate size reduction and Gauss reduction with an example.

**Example.** Let $B = \begin{pmatrix} 2.1 & 3 \\ 1 & 1 \end{pmatrix}$ be a basis matrix for a two-dimensional lattice $\mathcal{L}$.

**Size Reduction:** The Cholesky factorization of $B^\top B$ yields

$$R = \begin{pmatrix} 2.3259 & 3.1385 \\ 0 & 0.3869 \end{pmatrix}$$

Since $|r_{1,2}| = 3.1385 > |r_{1,1}/2| = 1.1629$, this basis is not size reduced. So we replace $v$ by $v = v - \mu u$ such that $\mu = \left\lfloor \frac{\langle v,u \rangle}{\|u\|^2} \right\rceil = \left\lfloor \frac{3.1385}{2.3259} \right\rceil = 1$. This translation leads to a new basis vector

$$v = \begin{pmatrix} 0.9 \\ 0 \end{pmatrix}.$$

The first basis vector is unchanged, i.e, $\tilde{u} = u$, hence $\tilde{r_{12}} = r_{12} - r_{11} = 0.8126 < |r_{11}|/2$. And finally the new basis is size reduced.

**Gauss Reduction:** After the size reduction, we notice that $\|\tilde{u}\| > \|\tilde{v}\|$. So, we perform a column swap $\tilde{u} \leftrightarrow \tilde{v}$, that leads to the basis matrix

$$\begin{pmatrix} 0.9 & 2.1 \\ 0 & 1 \end{pmatrix}$$

which is an upper triangular matrix. Since $|\tilde{r_{12}}| = 2.1 > |\tilde{r_{11}}|/2 = 0.4500$, we perform another size reduction step, i.e.,

$$v'' = \tilde{v} - \left\lfloor \frac{2.1}{0.9} \right\rceil \tilde{u} = \tilde{v} - 2.\tilde{u} = \begin{pmatrix} 2.1 \\ 1 \end{pmatrix} - 2. \begin{pmatrix} 0.9 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.3 \\ 1 \end{pmatrix}$$

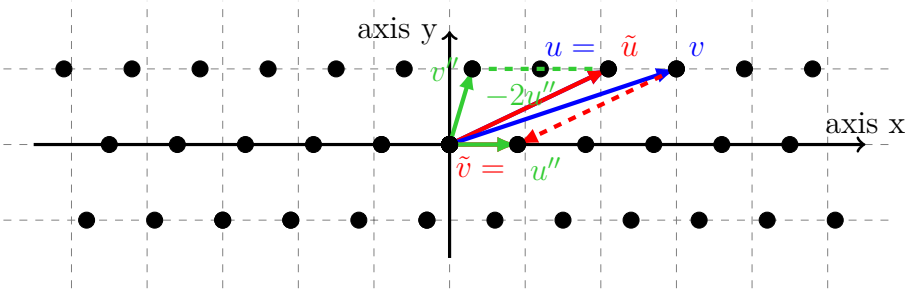


Figure 3.1: Illustration of size reduction (red) and Gauss reduction (green) for a two-dimensional lattice $\mathcal{L}$ spanned by the basis vectors $u = [2.1\ 1]^\top$ and $v = [3\ 1]^\top$ (shown in blue)

and

$$u'' = \tilde{v} = \begin{pmatrix} 0.9 \\ 0 \end{pmatrix}$$

Since $u''$ is shorter than $v''$, no further column swap or size reduction is possible and a Gauss reduced basis has been obtained i.e the basis

$$\begin{pmatrix} 0.9 & 0.3 \\ 0 & 1 \end{pmatrix}$$

consists of two shortest vectors that span $\mathcal{L}$.

The complexity of Gauss's algorithm is given by the following result:

**Theorem.** Given as input a basis $\{b_1, b_2\}$ of a lattice $\mathcal{L}$. Gauss's algorithm produces a lattice basis that realizes the successive minimum in time

$$O\left(\log \|b_2\| \,.\, [1 + \log \|b_2\| - \log \lambda_1(\mathcal{L})]\right).$$

The main difficulty is to prove that the total number of loop iterations is $O\left(1 + \log \|v\| - \log \lambda_1(\mathcal{L})\right)$, where $v$ is the initial second basis vector, for details, see [137].

## 3.5 Complexity of the LLL algorithm

Let us make some important observations on this reduction. We can see that as Gauss's algorithm, there are essentially two types of operations: translations and exchanging vectors. The aim of these translations is to

return the $\mu_{ij}$ coefficients with an absolute value lower than $\frac{1}{2}$. Thus letting $b_k$ closer to $b_k^*$ for making $b_k$ almost orthogonal to the previous vectors. Each of these translations is realized by a call to the **RED** procedure (see Algorithm. 3).

We take as an input a lattice basis $\{b_1, ..., b_m\}$ and output $\{b'_1, ..., b'_m\}$ a lattice basis of the same lattice. Let us suppose that, the first $k-1$ vectors $b_1, ..., b_{k-1}$ already satisfy the LLL conditions and that we have calculated the $\mu_{ij}$ coefficients for $j \leq i \leq k-1$ as well as $\|b_i^*\|^2$ for $1 \leq i \leq k-1$.
First, we take care of the first condition which is easy to satisfy. Indeed, suppose that we replace the $b_k$ vector by a vector of the form $b_k - qb_l$ with $l < k$ and $q$ an integer ( this step is to make sure that the projection of $b_k$ on $b_l^*$ for any $l < k$ is at most $\frac{1}{2}\|b_l^*\|$, and it does so by subtracting from the column $k$ some integer multiple of the column $l$ such that the $l$th coordinate becomes at most $\frac{1}{2}\|b_l^*\|$ in absolute value), then:

- The lattice generated by $b_1, ..., b_k$ is unchanged.
- The Gram-Schmidt orthogonalization vectors $b_1^*, ...., b_k^*$ stay also unchanged.
- $\mu_{kk}$,..., $\mu_{k,l+1}$ are unchanged. However, $\mu_{kl}$ is replaced by $\mu_{kl} - q$.

The last point is a consequence of the equality:

$$b_k - qb_l = \sum_{j=1}^{k} (\mu_{k,j} - q\mu_{l,j})\, b_j^*.$$

Therefore, we see that we can successively satisfy the conditions:
$|\mu_{k,k-1}| \leq \frac{1}{2}$ by replacing $b_k$ by $b_k - \lfloor \mu_{k,k-1} \rceil\, b_{k-1}$, then $|\mu_{k,k-2}| \leq \frac{1}{2}$ by replacing $b_k$ by $b_k - \lfloor \mu_{k,k-2} \rceil\, b_{k-2}$ and so on. To demonstrate this step, let us write $B$ in the orthonormal basis obtained by normalizing the Gram-Schmidt vectors

$$\begin{bmatrix} \|b_1^*\| & * & \cdots & \leq \frac{1}{2}\|b_1^*\| & & * \\ 0 & \|b_2^*\| & & & & \\ \vdots & & \ddots & & & \\ & & & \leq \frac{1}{2}\left\|b_{k-1}^*\right\| & & * \\ & & & \|b_k^*\| & & \\ & & & & \ddots & \\ 0 & \cdots & & & & \|b_m^*\| \end{bmatrix}$$

**RED**$(k, l)$ is the procedure which replaces $b_k$ by

$$b_k - \lfloor \mu_{k,l} \rceil\, b_l$$

and updates the $\mu_{kj}$ coefficients for $j \leq l$. So to obtain the condition (2.20), it is necessary to execute successively $\mathbf{RED}(k, k-1)$ until $\mathbf{RED}(k, 1)$. It should be noted that the condition (2.20) can be checked as soon as $\mathbf{RED}(k, k-1)$ is executed because the $\mu_{k,k-1}$ coefficient is not changed any more. Thus, we proceed in the following way: we execute $\mathbf{RED}\,(k, k-1)$, then later we test the second condition. If it is verified, we execute $\mathbf{RED}\,(k, k-2)$,....,$\mathbf{RED}\,(k, 1)$, and then we move to the next vector $b_{k+1}$. If it is not verified: we swap $b_k$ and $b_{k-1}$ and we go back to the previous case. The swaps, allow to decrease the length of $b^*_{k-1}$ at the expense of $b^*_k$, the aim is to shift the weight in $B^*$ from the first vectors to the last ones, allowing to minimize the first vectors and improve the basis quality.

Note that after this exchange, we have always a lattice basis. However, only the first $k-2$ vectors are LLL reduced. On the other hand, $b^*_{k-1}$ is modified. $\mathbf{SWAP}\,(k)$ is the procedure which exchanges the vectors $b_{k-1}$ and $b_k$ and updates $b^*_i$ and the $\mu_{ij}$ coefficients. The vectors $b^*_1, ..., b^*_{k-2}$ have not changed, nor $\mu_{ij}$ for $i \leq k-2$. As

$$b_k = b^*_k + \mu_{k,k-1} b^*_{k-1} + \mu_{k,k-2} b^*_{k-2} + ....$$

$b^*_{k-1}$ is replaced by $b^*_k + \mu_{k,k-1} b^*_{k-1}$ and the $\mu_{k-1,j}$'s coefficients by $\mu_{k,j}$ for $j < k-1$.

**Algorithm 3** RED

**Input**: $i$ and $j$ are integers.
**Output**: $|\mu_{ij}| \leq \frac{1}{2}$.
**if** $|\mu_{i,j}| > \frac{1}{2}$ **then**
  $r \leftarrow \lfloor \mu_{i,j} \rceil$ ;
  $b_i \leftarrow b_i - r b_j$;
  $\mu_{i,j} \leftarrow \mu_{i,j} - r$;
  $w \leftarrow 1$;
  **while** $w \leq l-1$ **do**
    $\mu_{i,w} \leftarrow \mu_{i,w} - r\mu_{j,w}$;
    $w \leftarrow w + 1$;
  **end while**
**end if**

In order to better understand the strategy of this reduction, we take the following simple example for $m = 2$:

**Example.** Let

$$B = \begin{pmatrix} 2.1 & 3 \\ 1 & 1 \end{pmatrix}$$

a lattice basis matrix. The Cholesky decomposition of $B^\top B$ yields the upper triangular matrix

$$R = \begin{pmatrix} 2.3259 & 3.1385 \\ 0 & 0.3869 \end{pmatrix}.$$

We notice from $R$ that the first condition (2.20) is not verified. So, we apply RED procedure for $k = 2$ and we obtain

$$v \leftarrow v - \lfloor 3.1385/2.3259 \rceil \, u = v - u = \begin{pmatrix} 0.9 \\ 0 \end{pmatrix},$$

and

$$\tilde{r_{12}} = r_{12} - r_{11} = 3.1385 - 2.3259 = 0.8126.$$

After the RED$(2, 1)$ procedure, we check the Lovász condition (2.21). For this, we take $\delta = 1/2$

$$\delta \, |\tilde{r_{11}}|^2 = 2.7049 > |\tilde{r_{22}}|^2 + |\tilde{r_{12}}|^2 = 0.8100.$$

According to the algorithm, a column swap, $u \leftrightarrow v$, must be done in this case, then we repeat what we had already done for $k = \max(k - 1, 2) = 2$. Thus, again

$$v = v - \lfloor 2.1/0.9 \rceil \, u = \begin{pmatrix} 2.1 \\ 1 \end{pmatrix} - 2 \begin{pmatrix} 0.9 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.3 \\ 1 \end{pmatrix}$$

and we obtain the following lattice basis matrix

$$\begin{pmatrix} 0.9 & 0.3 \\ 0 & 1 \end{pmatrix}$$

We notice that the second condition holds: $0.5.(0.9)^2 < 1^2 + (0.3)^2$. Since the second condition is verified, we move to the next case $k = 3$ but as $k = 3 > m = 2$, the algorithm terminates and the LLL reduced basis is

$$\begin{pmatrix} 0.9 & 0.3 \\ 0 & 1 \end{pmatrix}.$$

Note that a Gauss reduced basis is an LLL reduced basis.
A pseudo-code summarizes the main algorithmic steps (see, Algorithm. 4).

Now, it is clear that if the algorithm terminates, an LLL reduced basis lattice is obtained.

**Algorithm 4** LLL algorithm

**Input**: A lattice basis $[b_1, .., b_m]$.
**Output**: An LLL reduced basis.
Compute the Gram-Schmidt basis $b_1^*, ..., b_m^*$ and coefficients $\mu_{ij}$ for $1 \leq j < i \leq m$;
Compute $B_i = \langle b_i^*, b_i^* \rangle = \|b_i^\star\|^2$ for $1 \leq i \leq m$;
$k = 2$;
**while** $k \leq m$ **do**
  **for** $j = k - 1$ **to** 1 **do**
    Let $q_j = \lfloor \mu_{kj} \rceil$ and set $b_k = b_k - q_j b_j$;
    Update the values $\mu_{kj}$ for $1 \leq j < k$;
  **end for**
  **if** $B_k \geq \left(\delta - \mu_{k,k-1}^2\right) B_{k-1}$ **then**
    $k = k + 1$;
  **else**
    Swap $b_k$ with $b_{k-1}$;
    Update the values $b_k^*, b_{k-1}^*, B_k, B_{k-1}, \mu_{k-1,j}$ and $\mu_{kj}$ for $1 \leq j < k$, and $\mu_{ik}, \mu_{i,k-1}$ for $k < i \leq m$;
    $k = \max\{2, k - 1\}$;
  **end if**
**end while**

**Definition.** Let $B = \{b_1, ..., b_m\}$ be a lattice basis. The potential of $B$, denoted $\mathcal{D}_B$, is defined by

$$\prod_{i=1}^{m} \|b_i^*\|^{m-i+1} = \prod_{i=1}^{m} \|b_1^*\| \, \|b_2^*\| \, ... \, \|b_i^*\| = \prod_{i=1}^{m} \mathcal{D}_{B,i}.$$

where $\mathcal{D}_{B,i} := \det \Lambda_i$ and $\Lambda_i$ is defined as the lattice spanned by $b_1, .., b_i$.

We have already seen that during the Size Reduction step, the Gram-Schmidt basis does not change. Now consider the swap step. Suppose that $b_i$ is swapped with $b_{i+1}$. For all $k \neq i$, $\Lambda_k$ is not changed by the swap, and so $\mathcal{D}_{B,k}$. To prove this look at the two cases $k < i$ and $k > i$. When $k < i$ then there is no change in the basis $[b_1, ..., b_k]$, so the value of $\det(\mathcal{L}(b_1, ..., b_k))$ remains the same. On the other hand, if $k > i$ the only change is that two basis vectors in $[b_1, ..., b_k]$ are swapped, so the lattice $\mathcal{L}(b_1, ..., b_i)$ does not change and the determinant $\det(\mathcal{L}(b_1, ..., b_k))$ stays also the same. Therefore, only $\mathcal{L}(b_1, ..., b_i)$ is affected by the swap and only $\mathcal{D}_{B,i}$ changes. Consequently, let $\Lambda_i'$, $\mathcal{D}_{B,i}'$ denote the new values of $\Lambda_i$ and $\mathcal{D}_{B,i}$ respectively. We have that

$$\begin{aligned}
\frac{\mathcal{D}_{B,i}'}{\mathcal{D}_{B,i}} &= \frac{\det \Lambda_i'}{\det \Lambda_i} \\
&= \frac{\det \mathcal{L}(b_1, ..., b_{i-1}, b_{i+1})}{\det \mathcal{L}(b_1, .., b_i)} \\
&= \frac{\left(\prod_{j=1}^{i-1} \left\|b_j^*\right\|\right) \cdot \left\|\mu_{i+1,i} b_i^* + b_{i+1}^*\right\|}{\prod_{j=1}^{i} \left\|b_j^*\right\|} \\
&= \frac{\left\|\mu_{i+1,i} b_i^* + b_{i+1}^*\right\|}{\|b_i^*\|} \\
&< \sqrt{\delta}.
\end{aligned}$$

Where the last inequality follows from the condition in the swap step.
Thus each passage through Swap multiplies the product $\mathcal{D}_{B,1}\mathcal{D}_{B,2}...\mathcal{D}_{B,m}$ by a factor at most equal to $\sqrt{\delta}$. Therefore, we have just to show that the lower bound of this product is a constant depending only on the lattice.

**Proposition 5.** If $\mathcal{L}$ is a lattice of dimension $m$, then the quotient

$$\gamma(\mathcal{L}) := \frac{\min(\mathcal{L})}{\det(\mathcal{L})^{\frac{1}{m}}}.$$

is bounded from above by a constant depending only on $m$.

*Proof.* In fact, the quotient measures the density $\Delta$ of the spheres packing associated to a lattice $\mathcal{L}$. Let $r = \frac{\sqrt{\min(\mathcal{L})}}{2}$. The spheres centered at the lattice points of radius $r$ are interior disjoints, (it is the largest possible radius). Let $\Sigma$ be the set of these spheres and $\mathcal{V}$ the Voronoi region of the lattice $\mathcal{L}$, an important fundamental region, defined as the set of points in $\mathbb{R}^n$ that are closer to the origin than to any other point. The volume of this fundamental region is equal to $\sqrt{\det(\mathcal{L})}$. And the volume of $\mathcal{V} \cap \Sigma$ is equal to the volume of a sphere of radius $r$, i.e, $r^m \Pi_m$ where $\Pi_m$ is the volume of a sphere of radius 1 and dimension $m$, for details see [173], [33]. Then we have

$$\Delta = \frac{\mathrm{vol}(\mathcal{V} \cap \Sigma)}{\mathrm{vol}(\mathcal{V})} \leq 1.$$

Let

$$\frac{r^m \Pi_m}{\sqrt{\det(\mathcal{L})}} \leq 1, \text{ or } \left(\frac{\min(\mathcal{L})}{\det(\mathcal{L})^{\frac{1}{m}}}\right)^{\frac{m}{2}} \frac{\Pi_m}{2^m} \leq 1.$$

Then,

$$\gamma \leq \left(\frac{2^m}{\Pi_m}\right)^{\frac{2}{m}} = \frac{2^2}{\Pi_m^{\frac{1}{m}}}.$$

□

**Proposition.** The LLL algorithm terminates for arbitrary lattice bases.

*Proof.* The previous proposition. 5 shows:

$$\gamma_k := \mathrm{Sup}_{\mathcal{L} \subset \mathbb{R}^k} \gamma(\mathcal{L}), \text{ is finite.}$$

Therefore, we obtain for $\Lambda_k$,

$$\mathcal{D}_{B,k} \geq \left(\frac{\min(\Lambda_k)}{\gamma_k}\right)^k \geq \left(\frac{\min(\mathcal{L})}{\gamma_k}\right)^k.$$

And the product $\mathcal{D}_{B,1}....\mathcal{D}_{B,m}$ is well bounded from below by a constant and only depends on the lattice $\mathcal{L}$. Note that if $B$ was an integer basis, i.e, $B \in \mathbb{Z}^{n \times m}$, we can show that the LLL algorithm terminates in an easier way. As shown above, in each iteration, $\mathcal{D}_B$ decreases by a multiplicative factor, $\sqrt{\delta}$. Let $\mathcal{D}_{B,0}$ be the initial value of $\mathcal{D}_B$. Since $\mathcal{D}_B$ is a non zero integer, and in particular at least 1, this means that we can bound from above the number of iterations by

$$\log_{\frac{1}{\sqrt{\delta}}} \mathcal{D}_{B,0} = \frac{\log \mathcal{D}_{B,0}}{\log \frac{1}{\sqrt{\delta}}} \leq \frac{1}{\log \frac{1}{\sqrt{\delta}}} . \frac{m(m+1)}{2} \log\left(\max_i \|b_i\|\right).$$

(Since $\|b_i^*\| \leq \|b_i\|$, the initial value of $\mathcal{D}_B$ can be bounded from above by $(\max_i \|b_i\|)^{\frac{m(m+1)}{2}}$).
And this bound is for any constant $\delta < 1$. This proves an upper bound on the number of iterations as a function of the initial value of $\mathcal{D}_B$. Since $\mathcal{D}_B$ is computable in polynomial time from the input size, then its size must be polynomial in the input size.
We proved that the number of iterations is bounded by a polynomial in the input size. In order to mark the boundary of the running time of the algorithm, we still need to show that each iteration also takes polynomial time. It is not difficult to see that in each iteration we perform only a polynomial number of arithmetic operations (i.e., additions, multiplications, etc.) [103]. Hence, in order to prove a polynomial bound on the running time we only need to show that the size of the numbers involved in the entire computation also is polynomially bounded, see [89], [124], [103]. So, we have to prove that the bit length (i.e., the number of binary digits and it is necessary to represent an integer in the binary number system) of the numerators and denominators of the rational numbers $\|b_i^*\|^2$, $\mu_{i,j}$ and the bit length of the coefficients of the $b_i \in \mathbb{Z}^m$ are polynomially bounded throughout the algorithm. An analysis of the algorithm leads to a bound of the form $O(m \log \max_i \|b_i\|^2)$, [103].

The complexity of the LLL algorithm is traditionally assessed by counting the number of iterations, where an iteration is started whenever the Lovász condition is tested. In particular, when the Lovász condition is not satisfied. We had already given an upper bound for the number of iterations of LLL, which is polynomial in the data size, for all values of $\delta$ except the optimal value 1. This result depends on a quantity defined by

$$\mathcal{D} := \prod_{k=1}^{m-1} \prod_{l=1}^{k} |\tilde{r_{l,l}}|^2 ,$$

and which is bounded from below by 1 throughout the execution of the LLL algorithm when the basis matrix has an integer-valued. All of this implies that the number of swap operations carried out by the LLL algorithm is finite and the algorithm always terminates but under the condition that the original basis matrix $B$ is an integer-valued. A similar statement can be made also in the extreme case where $\delta = 1$, although the proof of this statement requires some additional work, see [37], [11].
The complexity analysis for the general case (real or complex valued) of $B$ is complicated by the fact that the quantity $\mathcal{D}$ is not necessarily $\geq 1$. However, other strictly positive lower (and upper) bounds on $\mathcal{D}$ can be found in this case. Based on such bounds, it was shown in [37] that the number of LLL

iterations in general is bounded from above by

$$m^2 \log_t \frac{A}{a} + m,$$

where $t = \frac{1}{\sqrt{\delta}}$, $A = \max_l |\tilde{r_{l,l}}|$ and $a = \min_l |\tilde{r_{l,l}}|$. This bound implies that the LLL algorithm terminates for arbitrary bases. Also this bound was used in [37] to prove polynomial average complexity of the LLL algorithm for special cases, for example, when the basis vectors are uniformly distributed inside the unit sphere in $\mathbb{R}^m$, and has been extended to i.i.d. (independent identically distributed) real and complex-valued Gaussian bases in the context of MIMO communications [107], [81]. □

## 3.6 Complexity of HKZ and Minkowski algorithms

**Algorithm 5** HKZ reduction

**Input**: A lattice basis $[b_1, .., b_m]$.
**Output**: An HKZ reduced basis.
1. Find the shortest lattice vector of the lattice, $\tilde{b}_1$.
2. Extend $\tilde{b}_1$ to a lattice basis.
3. Apply HKZ to $\left(\pi_2(\tilde{b}_2), ..., \pi_2(\tilde{b}_m)\right)$ where $\pi_2(.)$ be an orthogonal operator which projects "." onto $b_1^*$ where $b_1^*$ denotes the orthogonal complement of the subspace spanned by $b_1$.
4. Size reduce the obtained basis .

It is true, that Minkowski's and HKZ's reductions are the strongest but also the computationally most demanding to obtain. In both, the Minkowski and the HKZ lattice basis (see, definition 8, algorithm 5, definition 7) we notice that the first vector $b_1$ is the shortest vector in the lattice. This implies that the computation of the Minkowski and HKZ reduced bases are at least as complex as the computation of the shortest lattice vector. Computing an HKZ basis can be achieved by making $m$ calls to an SVP oracle (recursively find a shortest non zero lattice vector in the projected lattice). So, the two problems have the same time complexity up to a factor of $m$.
Zhang, Qiao and Wei, in [198], were the last to propose an algorithm for constructing an HKZ reduced basis. However, the expected complexity of their algorithm is exponential with the lattice dimension $m$. Therefore, no polynomial time algorithm in the dimension of the lattice is currently known for a HKZ reduced basis neither for a Minkowski reduced basis.

# Chapter 4

# Minkowski reduction algorithm

We are interested in lattices of low dimensions. It is desirable to have an algorithm to compute a "good" lattice bases quickly. However, the construction of such bases is a real problem.

Minkowski's reduction is the best one among all the existing lattice reduction algorithms, at least for low-dimensional lattice bases ($m \leq 4$). For this reason, we present our Minkowski reduction algorithm which is a modification of the Zhang, Qiao and Wei algorithm [198] and it is practical at least for lattices of low dimensions ($m \leq 5$).

Our algorithm simply applies the definition of Minkowski reduction (see, definition 7). This implies a basis that consists of shortest lattice vectors $m_i$ which can be extended to a basis of the lattice $\mathcal{L}$.
This work is divided into two parts where the first one is related to the shortest lattice vector problem (SVP). Since the enumerative algorithms were considered as the best among all the sphere decoding algorithms at least for low-dimensional lattice bases, we adopt here the Schnorr-Euchner enumeration with a little change in the way of updating the search radius. This algorithm updates the search radius when a shorter lattice vector satisfying the gcd constraint (i.e., $\gcd(z_{ii}, .., z_{im}) = 1$ where $z_i = [z_{i1}, ..., z_{im}] \in \mathbb{Z}^m$, for $i = 1, .., m$) is found (see, lemma 1).
The second part is about extending $\{m_1, ..., m_i\}$ to a basis for $\mathcal{L}$. It can be easily obtained by finding a unimodular matrix $Z$ such that $B_{i+1} = B_i Z$, where $B_i$ is a generator matrix of an $m$-dimensional lattice $\mathcal{L}$ such that the first $i-1$ columns of $B_i$ (i.e., $\{m_1, ..., m_{i-1}\}$) can be extended to a Minkowski reduced basis for $\mathcal{L}$.
We construct $Z$ such that the first $i-1$ columns of $B_{i+1}$ are the first $i-1$ columns $m_1$, $m_2$,..., $m_{i-1}$ and the $i$th column of $B_{i+1}$ is $m_i = B_i z_i$ where $z_i$ is the integer vector obtained by the Schnorr-Euchner enumeration for the $i$th position.

In this chapter, we present our algorithm for computing a Minkowski reduced basis up to dimension 5. We begin first with a detailed description of the algorithm then we show its drawbacks in section 4.1 . We compare the different notions of reduction by simple examples in section 4.2 and finally in section 4.3, we measure the quality of their output by what we call the orthogonality defect of lattice reduction algorithms.

## 4.1 A Descripition Of The Algorithm

We now state the following result useful for our algorithm.

**Lemma 2** ([187]). Let $B = [b_1, ..., b_m] \in \mathbb{R}^{n \times m}$ and $\mathcal{L}$ be the lattice generated by $B$. For a vector $v = \sum_{i=1}^{m} v_i b_i$, and any index $p$, $1 \leq p \leq m$, there exists a basis for $\mathcal{L}$ containing $\{b_1, ..., b_{p-1}, v\}$ if and only if $\gcd(v_p, ..., v_m) = 1$.

The goal of lattice reduction algorithms is to determine a sequence of elementary column operations (equivalently, a unimodular matrix $Z$) that transforms the given basis $B$ into a reduced basis $\tilde{B} = BZ$ according to the specific requirements of the corresponding reduction criterion.

The first Minkowski reduced basis vector $m_1$ is a shortest non zero lattice vector in $\mathcal{L}$, which can be obtained by applying Schnorr-Euchner enumeration [5], [159] and we can extend $m_1$ to a basis for $\mathcal{L}$ by calling Procedure **Transform** $(R, I_m, 1, z)$, (see, Algorithm 6).

Note that if the initial lattice basis $Y$ is a symmetric matrix then the upper triangular matrix $R = \text{Chol}(Y)$. Otherwise, $R = \text{Chol}(Y^\top Y)$.

### 4.1.1 Where This Idea Comes From?

Suppose that $m_1 = Bz$, where $z \in \mathbb{Z}^m$. Therefore, we have to construct an $m \times m$ unimodular matrix $Z$ whose first column is $z$. In other words, $Z^{-1}z = e_1$, which says that $Z^{-1}$, also unimodular, transforms $z$ into the first unit vector $e_1$.
For the special case when $m = 2$, we have the following algorithm:

**Procedure 1** (Unim2 $(p, q)$ [112]). Let $[p, q]^\top$ be a non zero integer vector and $\gcd(p, q) = d$. Using the **extended Euclidean** algorithm, we find integers $a$ and $b$ such that $ap + bq = d$. The integer matrix

$$M = \begin{bmatrix} p/d & -b \\ q/d & a \end{bmatrix}$$

**Algorithm 6** Transform $(R, Z, k, z)$ [112]

**Input**: given $R \in \mathbb{R}^{m\times m}$ upper triangular (obtained by Cholesky decomposition), $Z \in \mathbb{Z}^{m\times m}$ unimodular, and an integer vector $z \in \mathbb{Z}^{m-k+1}, 1 \leq k \leq m-1$, such that $\gcd(z_i) = \pm 1$.

**Output**: the updated $Z \in \mathbb{Z}^{m\times m}$ such that the integer vector $z$ is in the $k$th position.

**for** $j = m-k+1 : -1 : 2$ **do**
  $l = j + k - 1$;
  $z_1 = z(j-1); z_2 = z(j)$;
  $[d, a, b] = \gcd(z_1, z_2)$;
  $M = [[z_1/d; z_2/d], [-b; a]]$;
  $z(j-1) = d$;
  $R(1:l; l-1:l) = R(1:l, l-1:l) \times M$;
  $r = R(l-1:l, l-1)$;
  $r = r/\mathrm{norm}(r)$;
  $c = r(1); s = r(2)$;
  $G = [[c\ s]; [-s\ c]]$;
  $R(l-1:l, l-1:m) = G \times R(l-1:l, l-1:n)$;
  $Z(:, l-1:l) = Z(:, l-1:l) \times M$;
**end for**

is unimodular and

$$M^{-1}\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix},$$

such that

$$M^{-1} = \begin{bmatrix} a & b \\ -q/d & p/d \end{bmatrix}.$$

Thus, if $\gcd(p, q) = \pm 1$, then $z$ can be transformed by the unimodular matrix $M^{-1}$ into the first unit vector $e_1$.

Now, for the general case when $m > 2$. If $Bz$ is a shortest non zero lattice vector then by applying a sequence of plane unimodular transformations $\mathbf{M}$ as above to $z$, with $\gcd(z_i) = \pm 1$, $z$ can be transformed into the first unit vector.

Let us go back to lemma 2, suppose that $B_p$ is a generator matrix of an $m$-dim lattice $\mathcal{L}$ such that the first $p-1$ columns of $B_p$ can be extended to a Minkowski reduced basis for $\mathcal{L}$. Then it follows from lemma 2 that the $p$th Minkowski reduced basis vector $m_p$, which can be extended to an Minkowski

reduced basis with the first $p-1$ columns of $B_p$ must satisfy

$$\|\mathbf{m}_p\|_2 = \min\left\{\|\mathbf{B}_p z\|_2 : z \in \mathbb{Z}^m,\ \gcd(z_p, ..., z_m) = 1\right\}.$$

Obviously, the minimization problem can be considered as an SVP with the constraint $\gcd(z_p, ...., z_m) = 1$. Thus it is enough to incorporate the gcd constraint into the Schnorr-Euchner enumeration to solve it. A small change is going to take place in the way of updating the search radius.

Now, suppose that a basis $\{m_1, ..., m_{p-1}, b_p, ..., b_m\}$, $1 < p \leq m$, has been obtained, to extend $\{m_1, ..., m_{p-1}\}$ to a Minkowski reduced basis for $\mathcal{L}$, we have to solve the following two problems:

- Constructing the $p$th Minkowski reduced basis vector $m_p$.
- Extending $\{m_1, ..., m_p\}$ to a basis for $\mathcal{L}$.

We can use the length of the $p$th column as the initial size of the search region, so that at least one lattice point $z = e_p$, $(Rz_p = R(:, p))$ satisfying such a gcd constraint lies inside the search region. To further accelerate the search process, the LLL algorithm can be applied as a preprocessor.

**Algorithm 7** mdecode1 $(R, \omega, p)$

**Input**: $R \in \mathbb{R}^{m\times m}$, the LLL parameter $\omega$, and an index $p$, $1 \leq p \leq m$.
**Output**: a vector $z \in \mathbb{Z}^m$ such that $Rz$ is a shortest lattice point with $\gcd(z_p, ..., z_m) = 1$.
**if** $m = 1$ **then**
  return $z = 1$;
**end if**
set the initial size $r \leftarrow \|R(:, p)\|_2^2$;
utilize LLL algorithm to find a unimodular $Z$ and an upper triangular matrix $R_{new}$ such that $RZ$ is LLL reduced and $R_{new}$ is the $R$-factor of $RZ$;
$[z, \ell] \leftarrow$ msearch1 $(R_{new}, Z, 0, \emptyset, r, 0, p)$;

The msearch1 algorithm, (see, Algorithm 8), finds a solution for the closest vector problem (CVP) with the constraint $\gcd(z_p, ...., z_m) = 1$. We present now a recursive version (see, Algorithm 7, Algorithm 8) of this algorithm as Sph-Dec algorithm.

This algorithm is based on Schnorr-Euchner enumeration with the main difference between them on the way of updating the search radius. The original Schnorr-Euchner enumeration updates the search radius when a shorter lattice vector is found whereas the msearch1 algorithm updates the

**Algorithm 8** msearch1 $(R, Z, x, z_{in}, r, \text{dist}, p)$

**Input**: $R \in \mathbb{R}^{m\times m}$, a vector $x \in \mathbb{R}^m$ to decode, an integral partial solution $z_{in}$, the current distance record $r$, the distance between the current lattice vector and the corresponding sub-lattice, and an index $p$, $1 \le p \le m$.

**Output**: a vector $z \in \mathbb{Z}^m$ such that $RZ^{-1}z$ is a closest lattice point to $x$ satisfying $\gcd(z_p, .., z_m) = 1$, and $\ell = \|RZ^{-1}z - x\|_2^2$.

$\mathbf{LB} \leftarrow \left\lceil \frac{-\sqrt{r-\text{dist}}+x_m}{r_{m,m}} \right\rceil$, $\mathbf{UB} \leftarrow \left\lfloor \frac{\sqrt{r-\text{dist}}+x_m}{r_{m,m}} \right\rfloor$;

$\ell \leftarrow r$, $z \leftarrow \emptyset$;

**if** $\mathbf{LB} \le \mathbf{UB}$ **then**

  **for** each integer $s$ in the order of increasing distance from the center of $[\mathbf{LB}, \mathbf{UB}]$ **do**

    newdist $\leftarrow \text{dist} + (x_m - s.r_{m,m})^2$;

    **if** newdist $\le \ell$ **then**

      $\hat{z}_{in} \leftarrow [s; z_{in}]$;

      **if** $m > 1$ **then**

        $\hat{x} \leftarrow x(1 : m-1) - sR(1 : m-1, m)$;

        $[z', \ell'] \leftarrow \text{msearch1}(R_{m-1}, Z, \hat{x}, \hat{z}_{in}, \ell, \text{newdist}, p)$;

        **if** $\ell' \le \ell$ **then**

          set $\ell \leftarrow \ell'$, $z \leftarrow z'$;

        **end if**

      **else**

        **if** $\gcd(z_p, ..., z_m) = 1$ **then**

          set $\ell \leftarrow$ newdist;

          $z \leftarrow Z\hat{z}_{in}$;

        **end if**

      **end if**

    **else**

      return $z$ and $\ell$;

    **end if**

  **end for**

**end if**

search radius when a shorter lattice vector satisfying the gcd constraint is found.

After determining the $p$th Minkowski reduced basis vector $m_p = B_p z$, we move to our second problem: how can we extend $\{m_1, ...., m_p\}$ to a basis for $\mathcal{L}$? Simply, in terms of matrices, it is enough to find a unimodular matrix $Z$ such that

$$B_{p+1} = B_p Z,$$

which implies that the first $p-1$ columns of $Z$ are the first $p-1$ unit vectors $e_i$, $i = 1, ..., p-1$ and the $p$th column of $Z$ is the integer vector $z$ found by the mdecode1 algorithm, so that the first $p-1$ columns of $B_{p+1}$ equal the first $p-1$ columns $m_1, ..., m_{p-1}$ and the $p$th column of $B_{p+1}$ is $m_p = B_p z$ as desired. Since $\gcd(z_p, ..., z_m) = 1$, we can construct from procedure 1 a unimodular matrix $M_p$ whose first column is $[z_p, ..., z_m]^\top$. Now we consider the two $m \times m$ unimodular matrices

$$\mathbf{Z}_1 = \begin{bmatrix} \mathbf{I}_{p-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_p \end{bmatrix}, \quad \mathbf{Z}_2 = \begin{bmatrix} & z_1 & & \\ \mathbf{I}_{p-1} & \vdots & & \mathbf{0} \\ & z_{p-1} & & \\ & 1 & & \\ \mathbf{0} & & \ddots & \\ & & & 1 \end{bmatrix}$$

The product $\mathbf{Z}_1\mathbf{Z}_2$ is unimodular since both $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are unimodular and the first $p-1$ columns of $\mathbf{Z}_1\mathbf{Z}_2$ are the first $p-1$ unit vectors and the $p$th column of $\mathbf{Z}_1\mathbf{Z}_2$ is $z = [z_1, ..., z_m]^\top$. Therefore, the product $\mathbf{Z}_1\mathbf{Z}_2$ is an unimodular matrix that satisfies $B_{p+1} = B_p Z$. The application of $\mathbf{Z}_1$ can be performed by the Transform algorithm (see Algorithm 6) and the application of $\mathbf{Z}_2$ is the calculation of a linear combination of the first $p$ columns (see, $Z_1$ and $Z_2$ matrices). Putting all things together, the algorithm for constructing a Minkowski reduced basis of a lattice is $\text{mred}(B, \omega)$, see Algorithm 9.

Note that if $B$ is not a symmetric positive definite matrix, we apply Minkowski on $B$ and as mentioned before we take $R = \text{chol}(B^\top B)$ to finally obtain the symmetric definite positive matrix $Z^\top B^\top B Z$ where the latter is a Minkowski reduced matrix. Otherwise, we apply Minkowski on $R = \text{chol}(B)$ and we obtain at the end $Z^\top T^\top T Z$ a Minkowski reduced matrix.

**Algorithm 9** mred1 $(B, \omega)$

**Input**: $B \in \mathbb{R}^{n\times m}$, and the LLL parameter $\omega$, $\frac{1}{4} < \omega < 1$
**Output**: a unimodular $Z \in \mathbb{Z}^{m\times m}$ such that the columns of $BZ$ form a Minkowski reduced basis.
Cholesky factorization $R = \text{chol}(B^\top B)$;
$Z \leftarrow I_m$;
**for** $k = 1$ **to** $m$ **do**
  $z \leftarrow \text{mdecode1}\,(R, \omega, k)$;
  **if** $k = m$ **then**
    $Z = Z \times \left[\left[\mathbf{I}_{m-1}; \mathbf{0}^{1\times m-1}\right] z\right]$;
  **else**
    $[\,, \mathbf{Z}] \leftarrow \text{Transform}\,(R, Z, z, k)$;
    $Z(:, k) = Z(:, k) + Z(:, 1 : k-1)\,[z_1, ..., z_{k-1}]^\top$;
    $R = \text{chol}\left((BZ)^\top \times (BZ)\right)$;
  **end if**
**end for**

### 4.1.2 Why do we change every time the Transform function?

This algorithm will be changed including the algorithm named "Transform" (see, Algorithm 6), according to the size of the selected lattice for $p \geq 2$. So, as an example we take a 3-dimensional lattice generated by $\mathbf{Y}$.

$$\mathbf{Y} = \begin{bmatrix} 0.5603 & -0.2109 & 0.4901 \\ -0.2109 & 0.4309 & -0.0287 \\ 0.4901 & -0.0287 & 0.7630 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0.7486 & -0.2818 & 0.6546 \\ 0 & 0.5929 & 0.2628 \\ 0 & 0 & 0.5151 \end{bmatrix}$$

where $\mathbf{R} = \text{Chol}(\mathbf{Y})$.
First, by mdecode1 $(\mathbf{R}, 0.5, 1)$ (see, Algorithm 7), we obtain the shortest lattice vector $z_1'$,

$$z_1' = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} z_{11}' \\ z_{12}' \\ z_{13}' \end{bmatrix},$$

and by Transform $(1, z_1)$ (see, Algorithm 6), we extend $\mathbf{R}z_1'$ to a basis for $\mathcal{L}\,(R)$,i.e., $\mathbf{R}Z$.
For $j = 3$ and $l = 3$, we have $z_1 = z_{12}'$, $z_2 = z_{13}'$ and

$$M = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$z_2 = 1$,

$$\begin{aligned} Z(:, 2:3) &= Z(:, 2:3)M \\ &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} M \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix}. \end{aligned}$$

Now, for $j = 2$, $l = 2$ we have $z_1 = z'_{11}, z_2 = 1$

$$M = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}$$

and

$$\begin{aligned} Z(:, 1:2) &= Z(:, 1:2)M \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} M \\ &= \begin{bmatrix} 1 & -1 \\ 0 & 0 \\ -1 & 0 \end{bmatrix} \end{aligned}$$

Finally, for $p = 1$

$$Z = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}.$$

For $p = 2$. Again, by mdecode1$(\mathbf{R}, 0.5, 2)$, we obtain the second integer vector,

$$z_2 = \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix},$$

we have in this case $Z = Z_1 Z_2$, where

$$Z_1 = \begin{bmatrix} \mathbf{I}_1 & 0 \\ 0 & M_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & z_{22} & \star \\ 0 & z_{23} & \star \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & \star \\ 0 & -1 & \star \end{bmatrix}$$

and

$$Z_2 = \begin{bmatrix} 1 & z_{21} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

If we take a look at the Transform algorithm, we notice that $j = 3 - 2 + 1$ and $[d, a, b] = \gcd(z_{21}, z_{22})$. However, this should be $[d, a, b] = \gcd(z_{22}, z_{23})$ and

$$M = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Therefore,

$$Z = Z_1 Z_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}.$$

For the last $p$, $p = 3$, we have by mdecode1$(\mathbf{R}, 0.5, 3)$,

$$z_3 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

and

$$Z = \begin{bmatrix} 1 & 0 & z_{31} \\ 0 & 1 & z_{32} \\ 0 & 0 & z_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Finally, we multiply all the unimodular matrices obtained at the end of each case,

$$Z = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ 0 & -1 & 0 \\ -1 & 1 & 0 \end{bmatrix}.$$

Thus, by this $Z$, we form a new basis matrix for $\mathcal{L}(R)$ such that $Z^\top \mathbf{R}^\top \mathbf{R} Z = Z^\top \mathbf{Y} Z$ is Minkowski reduced.

Looking now at an example for $m = 4$

$$Y = \begin{bmatrix} 1.1134 & 0.2193 & -0.4414 & 0.0126 \\ 0.2193 & 0.8688 & 0.4175 & 0.2081 \\ -0.4414 & 0.4175 & 0.8351 & -0.0239 \\ 0.0126 & 0.2081 & -0.0239 & 0.6607 \end{bmatrix}$$

$$R = \begin{bmatrix} 1.0552 & 0.2078 & -0.4183 & 0.0119 \\ 0 & 0.9086 & 0.5552 & 0.2263 \\ 0 & 0 & 0.5932 & -0.2437 \\ 0 & 0 & 0 & 0.7416 \end{bmatrix}$$

For $p = 1$, by the mdecode1$(\mathbf{R}, 0.5, 1)$, we obtain our first integer vector $z_1$,

$$z_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

and by the "Transform$(1, z_1)$" function, we form our first unimodular matrix ,

$$Z = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

such that

$$Z^{-1}.z_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

For $p = 2$, the second integer vector is

$$z_2 = \begin{bmatrix} 0 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} z_{21} \\ z_{22} \\ z_{23} \\ z_{24} \end{bmatrix}.$$

Now, how can we form a lattice basis, or in a simple words, how can we form a unimodular matrix $Z$ such that the second integer vector is this $z_2$?

We had already shown that $Z = Z_1 Z_2$, where

$$Z_1 = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_3 \end{bmatrix}, \; Z_2 = \begin{bmatrix} 1 & z_{21} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and $M_3$ is a $3 \times 3$ matrix such that the first vector column must be

$$z_2' = \begin{bmatrix} z_{22} \\ z_{23} \\ z_{24} \end{bmatrix},$$

then

$$M_3 = \begin{bmatrix} 1 & \star & \star \\ -1 & \star & \star \\ -1 & \star & \star \end{bmatrix}.$$

In this case, we apply the Transform$(1, z_2')$ function.

Then we obtain

$$M_3 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}.$$

Therefore,

$$Z_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix} \text{ and } Z_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally,

$$Z = Z_1 Z_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

and this $Z$ verifies

$$Z^{-1} z_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

If we multiply this $Z$ by the previous one for $p = 1$, we obtain

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

Now, what happens in the coded algorithm?
For $p = 2$, we have

$$z_2 = \begin{bmatrix} 0 \\ 1 \\ -1 \\ -1 \end{bmatrix} = z$$

then by Transform$(2, z)$ algorithm, we obtain

$$Z = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \neq \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

The problem came from $z_1 = z(j-1)$, $z_2 = z(j)$ and $z(j-1) = d$. In the case $p = 2$, it has to be $z(j) = d$ and not $z(j-1) = d$, $z_1 = z(j)$ and $z_2 = z(j+1)$. There is no problem for $p = 3$ since $j = m - p + 1 = 4 - 3 + 1 = 2$. However, another problem appears in this case, $z_1$ and $z_2$ must be changed from $z(j-1)$ and $z(j)$ to $z(j+1)$ and $z(j+2)$.

The same problems arise if we take 5 as the dimension of a lattice. For $p = 2$, $z(j-1) = d$ must be replaced by $z(j) = d$ and $z_1$, $z_2$ should be replaced by $z(j)$ and $z(j+1)$. For $p = 3$, we should take $z_1 = z(j+1)$, $z_2 = z(j+2)$ and $z(j+1) = d$ and finally, for $p = 4$, $z_1 = z(j+2)$, $z_2 = z(j+3)$.

Sometimes, the $z$ obtained by the mcode1 algorithm have a particular shape and cause problems in the algorithm. That is why we had the idea of taking each $m$ independently in an algorithm.

For example if $p = 1$ and

$$z = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

then we should add in this case, if $z_1 = 0$ and $z_2 = 0$ then $j = j - 1$.

Note that after obtaining a Minkowski reduced basis, we have to verify Minkowski's conditions. Also in order that a simple real symmetric Minkowski reduced basis $M = (a_{i,j})_{1 \leq i,j \leq m}$ has $a_{j,j+1} \geq 0$ for all $j \in \{1, .., m-1\}$, an algorithm. 10 is also given, named **Sign**2 for $m = 2$ and **Sign**3 for $m = 3$ (see, Algorithm 11). We can also construct such an algorithm for arbitrary $m$ however because of the space occupied by the algorithm for $m > 3$, we deal only with dimensions 2 and 3.

Note that if $B$ is not a symmetric positive definite matrix, we apply Sign2 on $(B, Z)$.

We illustrate now the *simple Minkowski reduction* with an example. Let $Y$ be a symmetric positive definite matrix,

$$Y = \begin{bmatrix} 0.7106 & -0.8157 & 0.4323 \\ -0.8157 & 1.0491 & -0.6070 \\ 0.4323 & -0.6070 & 0.3989 \end{bmatrix}$$

and

$$R = \text{Chol(Y)} = \begin{bmatrix} 0.8430 & -0.9676 & 0.5128 \\ 0 & 0.3358 & -0.3299 \\ 0 & 0 & 0.1646 \end{bmatrix}$$

We start by calling the sphere decoding algorithm and hence, the first integer

**Algorithm 10** Sign2$(R, Z)$

**Input**: $B$ is a symmetric positive definite matrix, $R = \text{Chol}(B)$ an upper triangular matrix and $Z$ the unimodular matrix obtained by Minkowski reduction algorithm.
**Output**: $U$ an unimodular matrix such that $((RU)^\top(RU))_{i,i+1} \geq 0$.
$U = \emptyset$;
$C = (RZ)^\top(RZ)$;
**if** $C(1,2) < 0$ **then**
  $U = [-Z(:,1)\ Z(:,2)]$;
  $C = (RU)^\top(RU)$;
  $Z = U$;
  **if** $C(1,2) < 0$ **then**
    $U = [-Z(:,1)\ -Z(:,2)]$;
    $C = (RU)^\top(RU)$;
    $Z = U$;
  **end if**
**end if**
**if** $C(1,2) >= 0$ **then**
  $U = Z$;
**end if**

vector $z_1$ such that $Rz_1$ is the shortest lattice vector spanned by $R$ is obtained,

$$z_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

To construct a lattice basis such that $Rz_1$ is the first lattice vector, we call the Transform$(1, z_1)$ algorithm, (see, Algorithm 6). This algorithm gives us the following unimodular matrix $Z$ for $p = 1$:

$$Z = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

we go up to the next level $p = 2$. We start with an update of the upper triangular matrix $\mathbf{R}$, simply by applying Cholesky's decomposition to $(RZ)^\top(RZ)$.

Then, again by the sphere decoding algorithm, we find the second integer vector,

$$z_2 = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix}.$$

**Algorithm 11** Sign3$(R, Z)$

**Input**: $B$ is a symmetric positive definite matrix, $R = \mathrm{Chol}(B) \in \mathbb{R}^{m\times m}$ an upper triangular matrix, $Z \in \mathbb{Z}^{m\times m}$ an unimodular matrix obtained by Minkowski reduction algorithm.

**Output**: $U$ an unimodular matrix such that $((RU)^\top(RU))_{i,i+1} \geq 0$.

$U = \emptyset$;
$C = (RZ)^\top(RZ)$;
**if** $C(1,2) < 0$ || $C(2,3) < 0$ **then**
  $Z(:,1) = -Z(:,1)$;
  $U = [Z(:,1)Z(:,2)Z(:,3)]$;
  $Z = U$;
  $C = (RZ)^\top(RZ)$;
  **if** $C(1,2) < 0$ || $C(2,3) < 0$ **then**
    $Z(:,2) = -Z(:,2)$;
    $U = [-Z(:,1)Z(:,2)Z(:,3)]$;
    $Z = U$;
    $C = (RZ)^\top(RZ)$;
    **if** $C(1,2) < 0$ || $C(2,3) < 0$ **then**
      $Z(:,3) = -Z(:,3)$;
      $U = [-Z(:,1) - Z(:,2)Z(:,3)]$;
      $Z = U$;
      $C = (RZ)^\top(RZ)$;
      **if** $C(1,2) < 0$ || $C(2,3) < 0$ **then**
        $Z(:,1) = -Z(:,1)$;
        $Z(:,2) = -Z(:,2)$;
        $U = [Z(:,1)Z(:,2) - Z(:,3)]$;
        $Z = U$;
        $C = (RZ)^\top(RZ)$;
        **if** $C(1,2) < 0$ || $C(2,3) < 0$ **then**
          $Z(:,1) = -Z(:,1)$;
          $Z(:,3) = -Z(:,3)$;
          $U = [Z(:,1) - Z(:,2)Z(:,3)]$;
          $Z = U$;
          $C = (RZ)^\top(RZ)$;
          **if** $C(1,2) < 0$ || $C(2,3) < 0$ **then**
            $Z(:,2) = -Z(:,2)$;
            $Z(:,3) = -Z(:,3)$;
            $U = [-Z(:,1)Z(:,2)Z(:,3)]$;
            $Z = U$;
          **end if**
        **end if**
      **end if**
    **end if**
  **end if**
**end if**
**if** $C(1,2) < 0$ || $C(2,3) < 0$ **then**
  $U = Z$;
**end if**

To form a basis matrix by this lattice vector $Rz_2$ already obtained. We had shown before that the unimodular matrix $Z$ which transform the basis matrix to a new one had the following form, $Z = Z_1 Z_2$ where

$$Z_1 = \begin{bmatrix} I_1 & \\ & M_2 \end{bmatrix}$$

and

$$Z_2 = \begin{bmatrix} 1 & z_{21} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$M_2$ is an unimodular matrix whose first column is $[z_{22}, z_{23}]^\top$, thus,

$$M_2 = \begin{bmatrix} -2 & 1 \\ -1 & 0 \end{bmatrix}$$

and finally,

$$Z = Z_1 Z_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & -1 & 0 \end{bmatrix} . \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & -1 & 0 \end{bmatrix}.$$

Before moving to the final step, we update the upper triangular matrix $\mathbf{R}$. Finally, for $p = 3$,

$$Z = \begin{bmatrix} 1 & 0 & z_{31} \\ 0 & 1 & z_{32} \\ 0 & 0 & z_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Now, we multiply the unimodular matrices $Z$ obtained at the end of each $p$,

$$Z = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & -1 \\ 2 & 1 & 0 \end{bmatrix} . \begin{bmatrix} 1 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & -1 & 0 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 0 & 1 \end{bmatrix},$$

$$A = (a_{ij})_{1 \le i < j \le m} = (RZ)^\top (RZ) = \begin{bmatrix} 0.1130 & 0.0123 & 0.0284 \\ 0.0123 & 0.1283 & -0.0464 \\ 0.0284 & -0.0464 & 0.1778 \end{bmatrix}.$$

Since $a_{23} < 0$, we apply $U = \text{Sign3}(R, Z) = \begin{bmatrix} 1 & 1 & -1 \\ 2 & 1 & -1 \\ 2 & 0 & -1 \end{bmatrix}$. And finally,

$$(RU)^\top (RU) = \begin{bmatrix} 0.1130 & 0.0123 & -0.0284 \\ 0.0123 & 0.1283 & 0.0464 \\ -0.0284 & 0.0464 & 0.1778 \end{bmatrix}.$$

## 4.2 Comparison Between Reduction Algorithms

**HKZ and LLL:** Both, the HKZ reduced basis and the LLL reduced basis require size-reduction. The main difference between them is that for each trailing $(n-i+1)\times(n-i+1)$ sub-matrix of the upper triangular matrix $R$ of the lattice generating matrix, an HKZ reduced basis requires that its first column be a shortest non-zero vector in the lattice generated by the sub-matrix, while an LLL-reduced basis only requires that its first column scaled by a factor of $\delta$ be shorter than its second column. Thus, an HKZ reduced basis, is an LLL reduced basis for any $0.25 < \delta < 1$.

**HKZ and Minkowski:** In dimension 2, HKZ reduction is equivalent to Minkowski's reduction. But, HKZ and Minkowski reduction may differ from dimension 3 on. In particular, when a basis is Minkowski reduced, this basis must verify $\|b_1\| \leq .... \leq \|b_n\|$, which is not necessarily true for HKZ reduced bases. For this, we take as an example a lattice generated by $b_1 = [0.6410; 0.6698; -0.2927]$, $b_2 = [0.6698; 1.3165; -0.5127]$ and $b_3 = [-0.2927; -0.5127; 0.6450]$ where $B = \{b_1, b_2, b_3\}$ and the upper triangular matrix $R = \text{Chol}(Y)$.

As the previous reduction algorithm, we start by searching the first shortest lattice vector. Note that we adapt here the HKZ algorithm of Zhang, Qiao and Wei, [199].

By the sphere decoding algorithm, we obtain our first shortest vector

$$z_1 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$

Then by Transform$(1, z)$, we extend $z_1$ to a basis of the lattice and we obtain

$$Z = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

And

$$R = \text{chol}\left((RZ)^\top (RZ)\right) = \begin{bmatrix} 0.7861 & 0.0366 & -0.2799 \\ 0 & 0.7998 & -0.3532 \\ 0 & 0 & 0.6648 \end{bmatrix}$$

Now, to find a non zero vector $z_2 \in \mathbb{Z}^2$ such that $R(2:3, 2:3)z_2$ (where $R(2:3, 2:3)$ is a sub-matrix of $R$ formed by the second and the third rows

and the second and the third columns of $R$) is the shortest vector of the lattice generated by $R(2:3, 2:3)$, the sphere decoding algorithm is also used, and we obtain as a solution

$$z_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

As before, we apply the transform function, Transform$(2, z_2)$ and we obtain

$$M_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}.$$

We multiply the unimodular matrices $Z$ obtained at the end of each case

$$Z = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Now, we have to verify the size reduction condition:

$$R = \text{chol}((RZ)^\top (RZ)) = \begin{bmatrix} 0.7861 & 0.0367 & 0.2799 \\ 0 & 0.7998 & 0.3532 \\ 0 & 0 & 0.6648 \end{bmatrix}$$

$$r_{12} = 0.0367 < \frac{0.7861}{2}$$
$$r_{13} = 0.2799 < \frac{0.7861}{2}$$
$$r_{23} = 0.3532 < \frac{0.7998}{2}$$

So, the upper triangular matrix $R$ is size reduced and finally,

$$(RZ)^\top (RZ) = \begin{bmatrix} 0.6179 & 0.2200 & 0.0288 \\ 0.2200 & 0.6450 & 0.2927 \\ 0.0288 & 0.2927 & 0.6410 \end{bmatrix}.$$

We notice that $\left\|\tilde{b_2}\right\|^2 = 0.6450 > \left\|\tilde{b_3}\right\|^2 = 0.6410$. However, the Minkowski reduction gives us the following reduced matrix:

$$(RZ)^\top (RZ) = \begin{bmatrix} 0.6179 & 0.0288 & -0.2200 \\ 0.0288 & 0.6410 & -0.2927 \\ -0.2200 & -0.2927 & 0.6450 \end{bmatrix},$$

where

$$Z = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

As we know, for a basis to be reduced in the sense of HKZ, the lattice basis $B$ must verify the size reduction condition which is not the case for the Minkowski reduction.

Let

$$B = \begin{bmatrix} 100 & 49 & 0 \\ 0 & 100 & 62 \\ 0 & 0 & 100 \end{bmatrix}.$$

This lattice basis is Minkowski reduced, it suffices to check Minkowski's conditions to prove it. But it is not an HKZ reduced basis since $r_{23} = 62 > \frac{r_{22}}{2} = 50$.

**LLL and Minkowski:** We start first by examples to illustrate the difference between an LLL and a Minkowski reduced basis, and for this, we consider a symmetric real $4 \times 4$ matrix (for the ease of representation, we only give 4 digits)

```
Y=
0.7563   0.4850  0.4806  0.3846
0.4850   1.3631  0.2669 -0.3084
0.4806   0.2669  0.7784 -0.4523
0.3846  -0.3084 -0.4523  1.7538
```

This matrix was created using a $4 \times 4$ matrix $L$ with random entries, and then setting $Y = L^\top L$. As usual we put $Y = R^\top R$ where the upper triangular matrix $R$ is obtained from $Y$ via a Cholesky decomposition.

To this matrix $R$, we apply the algorithm for Minkowski reductions discussed above based on a successive finding of shortest lattice vectors. The found matrix $\tilde{Y} = \tilde{R}^\top \tilde{R}$ is then postprocessed to ensure the simple Minkowski reduction condition $\tilde{Y}_{i,i+1} \geq 0$, $i = 1, 2, 3$, i.e., a unimodular matrix $\tilde{Z}$ is constructed such that $\tilde{Z}^\top \tilde{Y} \tilde{Z}$ has positive elements in the right parallel to the diagonal. This leads to the matrix

```
 0.5321    0.2058   -0.1639    0.0181
 0.2058    0.5735    0.0920    0.2634
-0.1639    0.0920    0.5741    0.1364
 0.0181    0.2634    0.1364    0.6535.
```

In particular it can be seen that the squared length of the shortest lattice vector is 0.5321. An LLL reduction with $\delta = 3/4$ of the matrix $Y$ leads to

```
 0.7563   -0.2757    0.3182   -0.1089
-0.2757    0.5735    0.0920    0.2634
 0.3182    0.0920    0.5741    0.1364
-0.1089    0.2634    0.1364    0.6535.
```

The length of the shortest vector identified by the LLL algorithm is in this example 0.5735, thus longer than the shortest vector of the lattice which is still 0.5321 since both the LLL and the Minkowski reduction of a lattice are obtained via unimodular transformations. The latter obviously do not change the length of the shortest vector. Thus, this example shows that the LLL algorithm can lead to a considerable overestimation of the length of the shortest vector even for small size of the matrix. The effect is known to grow exponentially with the size of the matrix.

Note that in the above example, the shortest vector appears as the second vector in contrast to a Minkowski ordered matrix where the shortest vector is always the first.

**Remark.** An LLL reduced matrix is always ordered in accordance with the LLL condition. Thus there is no reason why the shortest vector should appear in the first position as in Minkowski reduced matrices. This is especially important in the context of the Siegel algorithm (see, Chapter 5) where the shortest vector is always assumed to be the first of the matrix.

To illustrate this aspect even more, we consider another example of a random matrix,

```
Y =

1.7472    0.5191    1.0260    0.6713
0.5191    1.3471    0.2216   -0.5122
1.0260    0.2216    0.6801    0.4419
0.6713   -0.5122    0.4419    0.7246.
```

The Minkowski reduction yields

```
0.2205    0.0443    0.0342    0.0351
0.0443    0.3636    0.1660   -0.0294
0.0342    0.1660    0.3688    0.1516
0.0351   -0.0294    0.1516    0.3753.
```

Table 4.1: Upper bounds for the orthogonality defect of HKZ, LLL ($\delta = 3/4$) and Minkowski reduced bases.

| m | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 24 |
|---|---|---|---|---|---|---|---|---|
| $\gamma_m$ | $2/\sqrt{3}$ | $2^{\frac{1}{3}}$ | $\sqrt{2}$ | $8^{\frac{1}{5}}$ | $(64/3)^{\frac{1}{6}}$ | $64^{\frac{1}{7}}$ | 2 | 4 |
| $\delta_{H,m}$ | 1.291 | 1.937 | 3.623 | 7.246 | 17.75 | 48.61 | 161.2 | $4.26 \times 10^{13}$ |
| $\delta_{L,m}$ | 1.414 | 2.828 | 8 | 32 | 181.0 | $1.45 \times 10^3$ | $1.64 \times 10^4$ | $3.48 \times 10^{41}$ |
| $\delta_{M,m}$ | 1.155 | 1.414 | 2 | 3.162 | 6.455 | 15.63 | 48.83 | $2.51 \times 10^{17}$ |

The corresponding LLL reduced matrix ($\delta = 3/4$) takes the form

```
 0.3753    0.0294   -0.1516    0.0351
 0.0294    0.3636    0.1660   -0.0443
-0.1516    0.1660    0.3688   -0.0342
 0.0351   -0.0443   -0.0342    0.2205.
```

In this case the shortest vector is found by the LLL algorithm in contrast to the previous example, but it appears as the last vector. The first vector has with 0.3753 almost twice the length of the shortest vector, 0.2205.

Note that the algorithm [38] implemented in Maple uses the LLL algorithm on $Y$ instead of an exact determination of the shortest lattice vector. As discussed above and illustrated below, this is considerably more rapid than an exact determination of the vector, but can lead to exponentially (in the dimension $g$) growing errors in this context.

## 4.3 The Orthogonality Defect Of Lattice Reduction Algorithms

We denote by $\delta_{H,m}, \delta_{L,m}$ and $\delta_{M,m}$ the upper bounds of the orthogonality defect over all $m \times m$ HKZ, LLL (with $\delta = 3/4$) and Minkowski reduced bases, respectively. Then from

$$\prod_{i=1}^{m} \|b_i\| \leq \left(\gamma_m^m \prod_{i=1}^{m} \frac{i+3}{4}\right)^{\frac{1}{2}} \mathrm{vol}(\mathcal{L}) \quad \text{(HKZ)}$$

$$\prod_{i=1}^{m} \|b_i\| \leq \beta^{\frac{m(m-1)}{4}} \mathrm{vol}(\mathcal{L}), \text{ where } \beta = (\delta - \frac{1}{4})^{-1} \quad \text{(LLL)}$$

$$\prod_{i=1}^{m} \|b_i\| \leq \gamma_m^{\frac{n}{2}} \left(\frac{5}{4}\right)^{\frac{(m-3)(m-4)}{4}} \mathrm{vol}(\mathcal{L}) \quad \text{(Minkowski)}$$

$$\gamma_m \leq 1 + \frac{m}{4}, \text{ for all } m \geq 1 \quad \text{(Hermite's constant)}.$$

We obtain,

$$\begin{aligned}
\delta_{H,m} &\leq \gamma_m^{\frac{m}{2}} \left( \prod_{i=1}^{m} \frac{i+3}{4} \right)^{\frac{1}{2}} = 2^{O(m \log m)} \\
\delta_{L,m} &\leq 2^{\frac{m(m-1)}{4}} \\
\delta_{M,m} &\leq \gamma_m^{\frac{m}{2}} \left( \frac{5}{4} \right)^{\frac{(m-3)(m-4)}{4}} = \left( \frac{5}{4} \right)^{\frac{m^2}{4} + O(m \log m)}.
\end{aligned}$$

We see from Table 4.1 that the upper bound of the orthogonality defect of Minkowski reduced bases is better (slightly smaller) than that of HKZ reduced bases. However, both are expected to be more orthogonal than LLL, especially for $m = 6, 7, 8$ and generally for lattices of dimensions $m \leq 8$. And we notice that for lattices of high dimension, an HKZ reduced basis is expected to be more orthogonal than a LLL reduced basis or a Minkowski reduced basis, such as $m = 24$, and the gap between HKZ reduced bases and LLL reduced bases gets larger quickly as the dimension increases.

# Chapter 5

# On the action of the Symplectic Group on the Siegel Upper Half Space

In his fundamental paper [168] Siegel introduced a special symmetric space $H^g$ of dimension $g$ which is called now the *the g-dimensional Siegel upper half space*. $H^1$ is the hyperbolic upper half plane. $H^g$ is formally defined as the subset of $g \times g$ complex symmetric matrices whose imaginary part is a positive definite matrix.
In fact, the origin of $H^g$ goes back to Riemann who defined the Riemann matrix $A \in H^g$ corresponding to a compact Riemann surface of genus $g$, endowed with a specific complex structure.
$Sp(2g, \mathbb{R})$ is the biholomorphism group of $H^g$. Of special interest is the lattice $\Gamma_g = Sp(2g, \mathbb{Z})$ which is called the *Siegel modular group*.
The modular group name is related to the fact that the points of the quotient space $\Gamma_1/H^1$ where $\Gamma_1 = SL(2, \mathbb{Z})$ are moduli (= parameters) for the isomorphism classes of elliptic curves over $\mathbb{C}$. To each point $\tau \in H^1$ one can associate the lattice $\Lambda_\tau = \mathbb{Z}\tau + \mathbb{Z}1 \subset \mathbb{C}$ and the quotient space $E_\tau = \mathbb{C}/\Lambda_\tau$, which is an elliptic curve, i.e., it is at the same time a complex curve and an abelian group. Conversely, every elliptic curve over $\mathbb{C}$ can be obtained in this way, but not uniquely: if $E$ is such a curve, then $E$ can be written as the quotient $\mathbb{C}/\Lambda$ for some lattice (discrete rank 2 subgroup) $\Lambda \subset \mathbb{C}$ which is unique up to "*homotheties*" $\Lambda \mapsto \lambda\Lambda$ with $\lambda \in \mathbb{C}^*$, and if we choose an oriented basis $(\omega_1, \omega_2)$ of $\Lambda$ (one with $\mathcal{I}(\omega_1/\omega_2) > 0$) and use $\lambda = \omega_2^{-1}$ for the homothety, then we see that $E \cong E_\tau$ for some $\tau \in H^1$, but choosing a different oriented basis replaces $\tau$ by $\gamma.\tau$ for some $\gamma \in \Gamma_1$. The quotient space $\Gamma_1/H^1$ is the simplest example of what is called a moduli space, i.e., an algebraic variety whose points classify isomorphism classes of other algebraic

varieties of some fixed type, (see [68, 196]).
In this chapter, we would like to introduce the action of Siegel's modular group on the Siegel upper half space, and study the fundamental domain for this action defined for all $g$.

**Definition 9.** Siegel's fundamental domain $F_g$ is the set of $\Omega = (\Omega_{ij}) \in H^g$ that satisfies the three following conditions:

1. $|\mathcal{R}(\Omega_{ij})| \leq \frac{1}{2}$ for all $i, j \in \{1, ..., g\}$;
2. $\mathcal{I}(\Omega)$ is in the fundamental region of Minkowski reductions;
3. $|\det(C\Omega + D)| \geq 1$ for all $C, D \in G_g$.

However, this action known as Siegel's fundamental domain seems difficult to obtain for genus greater than 2. For genus 1, we have the well known elliptic fundamental domain and genus 2 is due to Gottschling's work [63]. However, two points prevent us for constructing this fundamental domain for arbitrary $g$. First, the Minkowski fundamental domain appearing in the second condition of this domain Def 9 is only known for $g \leq 3$. For the moment, the case $g = 3$ is the most promising to study in this context.
Secondly, the third condition of Def 9, i.e., finding the finite number of classes $\{C, D\}$ of coprime symmetric pairs $C$, $D$ for which the absolute value of $\det(C\tau + D) \geq 1$. Even less is known about the third condition, it would be interesting to explore the matrices $C$ and $D$ of Def 9 as Gottschling did in genus 2 to compute $|\det(C\tau + D)|$ at least for an interesting set of these matrices (say for rank C = 1).
Siegel in [171] gave an algorithm to approximately reach the fundamental domain. But unfortunately, another problem appears, the shortest vector problem (SVP) for higher dimensions.
Siegel's modular group and Siegel's upper half space have many applications [13]. For example, *theta functions* [48, 42] are classically connected with Riemann surfaces and modular group. An important step in the efficient computation of multi-dimensional theta functions is the construction of appropriate symplectic transformations for a given Riemann matrix assuring a rapid convergence of the theta series. Siegel's algorithm [171] is used to approximately map the Riemann matrix to the Siegel fundamental domain. The shortest vector of the lattice generated by the Riemann matrix is identified exactly, and the algorithm ensures that its length is larger than $\sqrt{3}/2$. The approach is based on a previous algorithm by Deconninck et al. [38] using the LLL algorithm for lattice reductions. Here, the LLL algorithm is replaced

by exact Minkowski reductions for small genus and an exact identification of the shortest lattice vector for larger values of the genus.
A good introduction to the Siegel modular group and the Siegel modular forms can be found in Freitag's and Klingen's books, [55, 90].
In this work, we will need the following definition:

**Definition.** The action of a group $G$ on a topological space $X$ is said to be properly discontinuous if for any compact set $K \subset X$

$$K \cap g(K) = \emptyset,$$

except for a finite number of elements $g \in G$.

In this chapter we focus on Siegel's fundamental domain. In sections 5.1 and 5.2, we give the well known results for genus 1 and 2. In section 5.3, we define Siegel's fundamental domain for general $g$. In section 5.4, we aim at an exact identification of this domain in genus 3, so we present some results concerning rank $C = 1$. In section 5.5, we present Siegel's approximative reduction algorithm. We show the relation between Siegel's fundamental domain of symplectic transformations of Riemann matrices and the efficiency of computation of higher dimensional theta functions. We give in this context a different strategy than the existing algorithms illustrated by examples.

## 5.1 Siegel fundamental domain for general g

We first define notions which are the main objects studied in this section.

**Definition.** The following subset:

$$H^g := \left\{ z \in \mathbb{C}^{g \times g} \mid z^\top = z,\ \mathcal{I}(z) > 0 \right\},$$

is called the Siegel Upper Half Space of (degree or genus) $g$. Clearly, it is a generalization of the usual complex upper half plane

$$H := \{ z \in \mathbb{C} \mid \mathcal{I}(z) > 0 \} = H^1.$$

**Definition.** The subgroup of $GL(2g, \mathbb{R})$ defined as

$$K = \left\{ M \in M(2g, \mathbb{R}) : M^\top J M = J \right\}, \tag{5.1}$$

where $J = \begin{pmatrix} 0 & I_g \\ -I_g & 0 \end{pmatrix}$, is called the (real) symplectic group of degree $g$ and denoted by $Sp(2g, \mathbb{R})$.

Notice that $\det(J) = 1$, $J^2 = -I_{2g}$ and $J^\top = J^{-1} = -J$.
It is often useful for practical purposes to use block-matrix notation and to write $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ where the entries $A, B, C$ and $D$ are $g \times g$ matrices.
So $M$ is symplectic if and only if

$$A^\top C = C^\top A, \quad B^\top D = D^\top B, \quad A^\top D - C^\top B = I_g. \tag{5.2}$$

We notice also if $M$ is symplectic, $M^\top$ is symplectic too since:

$$MJM^\top = MJ(JM^{-1}J^{-1}) = -MM^{-1}(-J) = J. \tag{5.3}$$

And we can always write $M^\top \in Sp(2g, \mathbb{R})$ in block-matrix form

$$M^\top = \begin{pmatrix} A^\top & C^\top \\ B^\top & D^\top \end{pmatrix},$$

such that

$$AB^\top = BA^\top, \quad CD^\top = DC^\top, \quad AD^\top - BC^\top = I_g. \tag{5.4}$$

It follows that we have the equivalences

$$M \in Sp(2g, \mathbb{R}) \Leftrightarrow M^\top JM = J \Leftrightarrow MJM^\top = J.$$

We have also the following formula for the inverse of $M$:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} D^\top & -B^\top \\ -C^\top & A^\top \end{pmatrix}, \tag{5.5}$$

such that if $M$ is a symplectic matrix $M^{-1}$ is symplectic too since

$$(M^{-1})^\top JM^{-1} = -(MJM^\top)^{-1} = J.$$

Notice that in the case $g = 1$ the formula above (5.5) reduces to the familiar

$$M^{-1} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

which is true for every $2 \times 2$ matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ such that $\det(ad - bc) = 1$.
This implies that for $g = 1$, we have $Sp(2, \mathbb{Z}) = SL(2, \mathbb{Z})$.

An action of $Sp(2g,\mathbb{Z}) \subset Sp(2g,\mathbb{R})$, a discrete subgroup called "Siegel modular group", on $H^g$ can be defined

$$Sp(2g,\mathbb{Z}) \times H^g \longrightarrow H^g \tag{5.6}$$

$$(\gamma,\tau) \longrightarrow \gamma.\tau = \frac{A\tau + B}{C\tau + D}, \tag{5.7}$$

where $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$.

**Proposition.** The action of the symplectic group on the Siegel upper half plane is well defined, transitive and biholomorphic.

*Proof.* For each $\gamma \in Sp(2g,\mathbb{Z})$ and $\tau \in H^g$, the following two identities can be easily verified:

$$(A\tau + B)^\top (C\tau + D) - (C\tau + D)^\top (A\tau + B) = 0, \tag{5.8}$$

$$(A\tau + B)^\top \overline{(C\tau + D)} - (C\tau + D)^\top \overline{(A\tau + B)} = \tau - \overline{\tau} = 2i\mathcal{I}(\tau). \tag{5.9}$$

In order to be sure that the action is well defined, it is necessary first to show that $C\tau + D$ is invertible for each $\tau \in H^g$ and for each $\gamma \in Sp(2g,\mathbb{Z})$. If not, there would be a non zero vector $z \in \mathbb{C}^g$ such that $(C\tau + D)z = 0$. Then (5.9) would imply

$$z^\top (A\tau + B)^\top \overline{(C\tau + D)}\bar{z} - z^\top (C\tau + D)^\top \overline{(A\tau + B)}\bar{z} = 2iz^\top \mathcal{I}(\tau)\bar{z} = 0,$$

which is impossible since $\mathcal{I}(\tau) > 0$.
Now, we prove that $\gamma.\tau = (A\tau + B)(C\tau + D)^{-1} \in H^g$ for each $\tau \in H^g$ and for each $\gamma \in Sp(2g,\mathbb{Z})$.
Since $C\tau + D$ is invertible under this hypothesis, (5.8) is equivalent to $\gamma.\tau^{-1} = \gamma.\tau$. From this assertion and (5.9), we deduce:

$$\begin{aligned} \mathcal{I}(\gamma.\tau) &= \frac{1}{2i}\left[(\gamma.\tau)^\top - \overline{(\gamma.\tau)}\right], \\ &= \frac{1}{2i}((C\tau + D)^{-1})^\top \ (5.8)\overline{(C\tau + D)^{-1}}, \\ &= ((C\tau + D)^{-1})^\top \mathcal{I}(\tau)(\overline{(C\tau + D)})^{-1}. \end{aligned}$$

Since $\tau \in H^g$, it follows that $\mathcal{I}(\gamma.\tau) > 0$.
Therefore, it has been shown that $\gamma.\tau$ is contained in $H^g$. We can immediately verify that $I_{2g}.\tau = \tau$ for each $\tau \in H^g$ and if $\gamma_1$, $\gamma_2$ belong to $Sp(2g,\mathbb{Z})$, then

$\gamma_1.(\gamma_2.\tau) = (\gamma_1\gamma_2).\tau$, for each $\tau \in H^g$. Hence, the action is well defined.
To prove transitivity, it is enough to find a symplectic map that transforms $\tau = iI$ to any $X + iY \in H^g$, $Y > 0$. Hence, we consider the composition of two symplectic maps associated with the following symplectic matrices

$$\nu = \begin{pmatrix} \sqrt{Y} & 0 \\ 0 & \sqrt{Y^{-1}} \end{pmatrix} \text{ and } \xi = \begin{pmatrix} I & X \\ 0 & I \end{pmatrix}$$

such that $(\xi\nu).\tau = X + iY$.
This action is biholomorphic since it is a rational function and the inverse of this action is obtained by acting on $\gamma^{-1}$, i.e., $\gamma^{-1}.\tau = \frac{D^\top\tau - B^\top}{-C^\top\tau + A^\top}$ where it is a rational function, too. □

$Sp(2g, \mathbb{Z})$ acts properly discontinuously on $H^g$. For example, for $g = 1$, given $\gamma \in SL(2, \mathbb{Z})$, we have $\gamma(F) \cap F = \emptyset$ unless $\gamma$ lies in a finite set of elements of $SL(2, \mathbb{Z})$ which have the fix point $\eta = \frac{1}{2} + i\frac{\sqrt{3}}{2} \in H^1$ and $i \in H^1$ where $F$ is the fundamental domain for the action of $SL(2, \mathbb{Z})$ on the upper half plane $H^1$.

Note that $-I_{2g}$ acts trivially on $H^g$, i.e, $-I_{2g}.\tau = \tau$ for each $\tau \in H^g$. Therefore, we can consider the action of

$$G_g := Sp(2g, \mathbb{Z}) / \langle \pm I_{2g} \rangle$$

called " the projective symplectic group", on $H^g$, where

$$\Gamma_g := Sp(2g, \mathbb{Z})$$

is called "the modular group".

**Proposition** ([133])**.** The modular group $\Gamma_g$ is generated by the three following classes of generators:

$$\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \begin{pmatrix} A & 0 \\ 0 & (A^{-1})^\top \end{pmatrix}, \begin{pmatrix} I & B \\ 0 & I \end{pmatrix}$$

for all $A \in GL(g, \mathbb{Z})$ and $B$ a symmetric, integer matrix.

*Proof.* See, [[133], proposition $A5$, p.210]. □

Siegel in [171] gave, for all $g$, the following fundamental domain for the action of $G$ on $H^g$.

**Definition 10.** Siegel's fundamental domain $F_g$ is the set of $\Omega = (\Omega_{ij}) \in H^g$ that satisfies the three following conditions:

1. $|\mathcal{R}(\Omega_{ij})| \leq \frac{1}{2}$ for all $i, j \in \{1, ..., g\}$;

2. $\mathcal{I}(\Omega)$ is in the fundamental region of Minkowski reductions;

3. $|\det(C\Omega + D)| \geq 1$ for all $C, D \in G_g$.

We continue to use the matrix representation of $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ in $Sp(2g, \mathbb{Z})$ to denote elements of $G_g$.

Note that the third condition in Def 10 can be seen as a condition of maximal height or a highest-point condition, i.e, you choose $\gamma \in \Gamma_g$ to maximize $\mathcal{I}(\gamma.\tau)$, see [90]. It must be satisfied for all the $\Gamma_g / \langle \pm I_{2g} \rangle$ matrices. However, it is shown that it is enough for this condition to be satisfied for a certain finite set, for all $g$ see [90], which is only known for $g = 1$ ($\{S\} = \left\{ \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \right\}$) and $g = 2$ (19 matrices due to Gottschling [63]).

**Remark.** For $g = 2$, the Minkowski fundamental region, in Def 10. 2, i.e the fundamental domain of the unimodular group, is given by the simple Minkowski reduction. However, for $g > 2$ the simple Minkowski reduction does not define the Minkowski fundamental domain, see [176]. The fundamental domain for $g = 3$ is given in [176] and chapter. 2, but the corresponding conditions in higher dimension appear to be unknown.

These three conditions address different parts of $G_g = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$. The first condition Def 10. 1 fixes the matrix $B$ in $G_g$. The second one Def 10. 2 refers to Minkowski reductions, see. chapter 2 and 4, and [121, 122] and fixes the unimodular matrix $A$ in $G_g$. The third and last one Def 10. 3 fixes the matrices $C, D$ in $G_g$.

**Proposition.** $F_g$ is a fundamental domain for the action of $G_g$ on $H^g$. Then for all $\Omega \in H^g$, there exists $\gamma \in G_g$ such that $\gamma.\Omega \in F_g$ and this element $\gamma$ is unique if $\gamma.\Omega$ is an interior point of $F_g$.

*Proof.* See, [[90], theorem 2 p.34], [168]. □

**Lemma 3.** For all $\Omega \in F_g$, let $\Omega_{11}$ be the first diagonal element of $\Omega$, then

$$\mathcal{I}(\Omega_{11}) \geq \frac{\sqrt{3}}{2}.$$

*Proof.* Let $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in G_g$ such that

$$A = \begin{pmatrix} 0 & 0_{g-1}^\top \\ 0_{g-1} & I_{g-1} \end{pmatrix}, \; B = \begin{pmatrix} -1 & 0_{g-1}^\top \\ 0_{g-1} & 0_{g-1,g-1} \end{pmatrix}, \tag{5.10}$$

$$C = \begin{pmatrix} 1 & 0_{g-1}^\top \\ 0_{g-1} & 0_{g-1,g-1} \end{pmatrix}, \; D = \begin{pmatrix} 0 & 0_{g-1}^\top \\ 0_{g-1} & I_{g-1} \end{pmatrix}, \tag{5.11}$$

where $0_{g-1}$ denotes the column vector of $g-1$ zeros, we notice that

$$|\det(C\Omega + D)| = |\Omega_{11}|.$$

Since $\Omega \in F_g$, we have $|\Omega_{11}| \geq 1$ and $|\mathcal{R}(\Omega_{11})| \leq \frac{1}{2}$ which implies that $\mathcal{I}(\Omega_{11}) \geq \frac{\sqrt{3}}{2}$. □

Note that since $\mathcal{I}(\Omega)$ is Minkowski reduced, then

$$\frac{\sqrt{3}}{2} \leq \mathcal{I}(\Omega_{11}) \leq \mathcal{I}(\Omega_{22}) ... \leq \mathcal{I}(\Omega_{gg}).$$

## 5.2 Genus 1

An early study of this action was done by C. L. Siegel in his 1943 book "Symplectic Geometry" [170].
$H^g$ and $\Gamma_g$ are meaningful generalizations of the usual upper half plane

$$H = H^1 = \{\tau \in \mathbb{C} \mid \mathcal{I}(\tau) > 0\},$$

and the linear fractional transformation group $SL(2,\mathbb{R})$ acting on $H$, where

$$SL(2,\mathbb{R}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid a,b,c,d \in \mathbb{R}, ad - bc = 1 \right\}.$$

The special linear group acts on $H$ via the "Möbius transformation" (5.7). This action is well defined and

$$\mathcal{I}(\gamma.\tau) = \frac{\mathcal{I}(bc\bar{\tau} + ad\tau)}{|c\tau + d|^2} = \frac{\mathcal{I}(\tau)}{|c\tau + d|^2}. \tag{5.12}$$

**Definition.** The projective symplectic group (or elliptic projective symplectic group) is defined by

$$G_1 = G := \Gamma_1 / \langle \pm I_2 \rangle ,$$

where $\Gamma_1 = SL(2, \mathbb{Z})$

It is just the image of $SL(2, \mathbb{Z})$ on

$$PSL(2, \mathbb{R}) := SL(2, \mathbb{R}) / \langle \pm I_2 \rangle ,$$

where $PSL(2, \mathbb{R})$ acts faithfully on $H$, i.e., for all $\tau \in H$, $\gamma.\tau = \tau \Rightarrow \gamma = e$.

We will show now the fundamental domain $F_1$ for the action of $G$ on $H$ and some important properties.

**Definition.** $G = \overline{\Gamma_1}$ is generated by the two elements

$$T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ and } S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

with the relations $S^2 = (ST)^3 = -I_2$. And the action of $S$ and $T$ on $H$ are given by

$$S : \tau \to -\frac{1}{\tau}, \quad T : \tau \to \tau + 1.$$

**Proposition.** The set

$$F_1 := \left\{ \tau \in H \mid |\tau| > 1, |\mathcal{R}(\tau)| < \frac{1}{2} \right\},$$

is a fundamental domain for the action of $G$ on $H$ (sometimes called, a fundamental domain for the full modular group $\Gamma_1$), see. Figure. 5.1.

**Remark.** For genus 1, it is clear that the third condition of Siegel's fundamental domain is $|\tau| > 1$ and this corresponds to the set $\{S\}$.

**Remark 1.** Note that the fundamental domain for $G_1$, is an open subset $F_1 \subset H$ such that no two distinct points of $F_1$ are equivalent under the action of $G_1$ and every point $\tau \in H$ is $G_1$-equivalent to some points in the closure $\overline{F_1}$ of $F_1$.
The points on the two lines $\mathcal{R}(\tau) = \pm\frac{1}{2}$ are equivalent under the action of

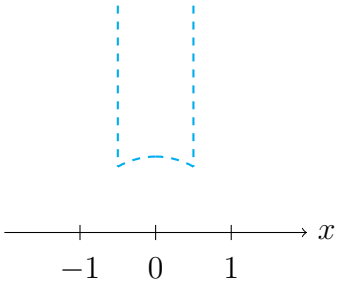


Figure 5.1: $F_1$

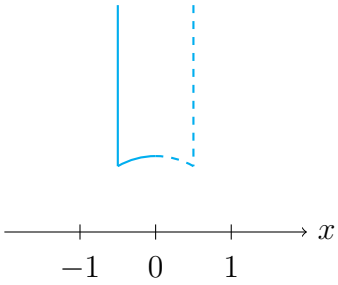


Figure 5.2: $\tilde{F_1}$

$T : \tau \to \tau \pm 1$. And the points on the left and right halves of the arc $|\tau| = 1$ are also equivalent under the action of $S : \tau \to -\frac{1}{\tau}$. These are the only equivalences for the points on the boundary. Therefore, $\tilde{F_1}$ is defined as the semi-closure of $F_1$ where only the boundary points with non-positive real part (see, Figure. 5.2) are added. Then every point of $H$ is $G_1$-equivalent to a unique point of $\tilde{F_1}$, i.e., $\tilde{F_1}$ is a strict fundamental domain for the action of $G_1$. However, most people use the words "fundamental domain" for the strict fundamental domain or for its closure rather, then for the interior (see, [196]).

*Proof.* Let $\tau \in H$. Then $\{m\tau + n \mid m, n \in \mathbb{Z}\}$ is a lattice in $\mathbb{C}$. We know that every lattice has a point different from the origin of minimal modulus. Let us suppose $c\tau + d$ be such a point. The integers $c, d$ must be relatively prime (otherwise we could divide $c\tau + d$ by an integer to get a new point in the lattice of even smaller modulus). So, there are integers $a$ and $b$ such that $\gamma_1 = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in G_1$. By (5.12) we deduce that $\mathcal{I}(\gamma_1.\tau)$ is a maximal member of the set $\{\mathcal{I}(\gamma.\tau) \mid \gamma \in G_1\}$. Now, we choose an $n$ such that $T^n(\gamma.\tau)$ is shifted into the vertical strip between $\frac{-1}{2}$ and $\frac{1}{2}$. Then $\tilde{\tau} = T^n(\gamma.\tau) \in \overline{F_1}$,

since if not, i.e., $|\tilde{\tau}| < 1$, we would obtain

$$\mathcal{I}(S\tilde{\tau}) = \mathcal{I}(-1/\tilde{\tau}) = \frac{\mathcal{I}(\tilde{\tau})}{|\tilde{\tau}|^2} > \mathcal{I}(\tilde{\tau}),$$

a larger imaginary part than $\tilde{\tau}$, contradicting the maximality of the imaginary part of $\gamma.\tau$. Therefore, for every $\tau \in H$ there exists $\gamma \in G_1$ such that $\gamma.\tau \in \overline{F_1}$. Suppose that given $\gamma \in G_1$, $\tau_1 = \tau \in F_1$ and $\tau_2 = \gamma.\tau \in F_1$ as well with $\gamma \neq \pm I_2$. As the pairs $(\tau, \gamma)$ and $(\gamma.\tau, \gamma^{-1})$ play symmetric roles here. We assume without loss of generality that $\mathcal{I}(\gamma.\tau) \geq \mathcal{I}(\tau)$ which implies from (5.12) that $|c\tau + d| \leq 1$. $\gamma$ cannot be of the form $T^n$ since this would contradict the condition $|\mathcal{R}(\tau_1)|$, $|\mathcal{R}(\tau_2)| < \frac{1}{2}$, so $\gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ with $c \neq 0$. $\mathcal{I}(\tau) > \frac{\sqrt{3}}{2}$ for all $\tau \in F_1$. Hence, from (5.12) we get

$$\frac{\sqrt{3}}{2} < \mathcal{I}(\tau_2) = \frac{\mathcal{I}(\tau_1)}{|c\tau_1 + d|^2} \leq \frac{\mathcal{I}(\tau_1)}{c^2\mathcal{I}(\tau_1)^2} < \frac{2}{c^2\sqrt{3}},$$

which can only be satisfied if $c = \pm 1$. However, $|\pm\tau_1 + d| \geq |\tau_1| > 1$, and this gives a contradiction with $\mathcal{I}(\gamma.\tau) \geq \mathcal{I}(\tau)$.

Given now two points $\tau_1 = \tau$, $\tau_2 = \gamma.\tau \in \overline{F_1}$, then either $\mathcal{R}(\tau_2) = \pm\frac{1}{2}$ and $\tau_2 = \tau_1 \pm 1$ or $|\tau_2| = 1$ and $\tau_2 = -\frac{1}{\tau_1}$, (we show Remark. 1). Indeed, $\mathcal{I}(\gamma.\tau) \geq \mathcal{I}(\tau)$ implies that $|c\tau + d| \leq 1$ which is impossible for $|c| \geq 2$. And consequently, we get $c \in \{-1, 0, 1\}$.

If $c = 0$:

we have $a = d = \pm 1$. Since $bc = 0$, $\gamma$ can be written as a translation of $\pm b$. Therefore, $Y = \begin{pmatrix} \pm 1 & \pm b \\ 0 & \pm 1 \end{pmatrix}$ which implies that $\gamma.\tau = \tau \pm b$.
$\tau_1$, $\tau_2 \in \overline{F_1}$, then $|\mathcal{R}(\tau_1)|$, $|\mathcal{R}(\tau_2)| \leq \frac{1}{2}$. From here, we deduce the following two cases:

1. $b = 0$ and then $\gamma = \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix}$,

2. $b = \pm 1$ and then $\mathcal{R}(\tau_2) = -\frac{1}{2}$ and $\mathcal{R}(\tau_1) = \frac{1}{2}$ (or vice-versa).

If $c = 1$:

$d = 0$ except for $\tau = \exp(\frac{2\pi i}{3})$ and $\tau = \exp(\frac{\pi i}{3})$ where $d = \{0, 1\}$, respectively $\{-1, 0\}$ which will be discussed in detail later.
It follows that $b = -1$ and $\gamma.\tau = \frac{a\tau+b}{c\tau+d} = a - \frac{1}{\tau}$. But since $\tau$ and $\gamma.\tau$ belong to $\overline{F_1}$. So, for every $\tau$ such that $\mathcal{R}(\tau) \neq \pm\frac{1}{2}$, we have $a = 0$ otherwise the translation would leave the fundamental domain.

If $c = -1$:

$d = 0$, it follows that $b = 1$ and $\gamma.\tau = -a - \frac{1}{\tau}$. But since $\tau$ and $\gamma.\tau$ belong to $\overline{F_1}$. So, again for every $\tau$, we have $a = 0$ otherwise the translation would leave the fundamental domain. □

**Proposition.** To each $\tau \in \overline{F_1}$, let Stabilizer$(\tau) = \{\gamma \in G_1 \mid \gamma.\tau = \tau\}$. Then Stabilizer$(\tau) = \{\pm I_2\}$ for all $\tau \in \overline{F_1}$ unless

- $\tau = i$, then Stabilizer$(\tau) = \langle S \rangle$ of order 2,
- $\tau = \exp(\frac{2\pi i}{3})$, then Stabilizer$(\tau) = \langle ST \rangle$ of order 3,
- $\tau = \exp(\frac{\pi i}{3})$, then Stabilizer$(\tau) = \langle TS \rangle$ of order 3.

*Proof.* We will show first why $\{0, 1\}$ respectively, $\{-1, 0\}$ are the only values for which $d$ can be taken for $\tau = \exp(\frac{2\pi i}{3})$ respectively, $\tau = \exp(\frac{\pi i}{3})$.

For $\tau = \exp(\frac{2\pi i}{3})$, $|\tau| = 1$ and $Re(\tau) = -\frac{1}{2}$. If $c\tau + d = \tau + 1$, we get

$$\mathcal{R}(c\tau + d) = \mathcal{R}(\tau + 1) = \mathcal{R}\left(-\frac{1}{2} + i\frac{\sqrt{3}}{2} + 1\right) = \mathcal{R}\left(-\frac{1}{2} + 1\right) = \mathcal{R}\left(\frac{1}{2}\right).$$

More precisely, this gives us $\exp(\frac{2\pi i}{3}) + 1 = \exp(\frac{\pi i}{3})$. Therefore $|\tau| = |\tau + 1| = 1$, i.e., $|c\tau + d| \leq 1$ when $d \in \{0, 1\}$.

For $\tau = \exp(\frac{\pi i}{3})$, this case is similar to the previous one where $d \in \{-1, 0\}$.

If $d = 0$:

we get $\gamma.\tau = a - \frac{1}{\tau}$ ($c = 1$, then $b = -1$). Hence, for $\tau = \exp(\frac{2\pi i}{3})$, we have

$$a - \frac{1}{\tau} = a + \frac{1}{-\exp(\frac{2\pi i}{3})} = a + \frac{1}{\exp(\frac{-i\pi}{3})} = a + \exp\left(\frac{i\pi}{3}\right).$$

Therefore, $a \in \{-1, 0\}$.

For $\tau = \exp(\frac{\pi i}{3})$, we apply the same strategy and we get $a \in \{0, 1\}$.

We obtain the identity, i.e., $\gamma.\tau = \tau$ in these two cases:

- $\gamma = \begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix} = (ST)^2$, for $\tau = \exp(\frac{2\pi i}{3})$,
- $\gamma = \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix} = (TS)$, for $\tau = \exp(\frac{\pi i}{3})$.

If $d = 1$:

for $\tau = \exp(\frac{2\pi i}{3})$. To determine the values of $a$, we have:

$$\gamma.\tau = a - \frac{1}{\exp(\frac{2\pi i}{3}) + 1} = a - \frac{1}{\exp(\frac{\pi i}{3})} = a + \frac{1}{\exp(-\frac{2\pi i}{3})} = a + \exp\left(\frac{2\pi i}{3}\right),$$

which implies that $a \in \{0, 1\}$.
Similarly, we show that for $d = -1$ and $\tau = \exp(\frac{\pi i}{3})$, the values of $a$ are $\{-1, 0\}$.

Therefore, the identity is obtained:

- if $\gamma = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix} = (ST)$, for $\tau = \exp(\frac{2\pi i}{3})$,
- if $\gamma = \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix} = (TS)^2$, for $\tau = \exp(\frac{\pi i}{3})$.

If $c = -1$: we return to $c = 1$ by simply changing the sign of the coefficients $a, b, c, d$.

Finally, for $\tau = i$, we have $S\tau = -\frac{1}{i} = i$, and therefore, we get the identity when $\gamma = S$. To summarize, we have that Stabilizer($\exp(\frac{2\pi i}{3})$) (respectively, Stabilizer($\exp(\frac{\pi i}{3})$), Stabilizer($i$)) is generated by $ST$, (respectively, $TS, S$). □

We deduce the following corollary:

**Corollary 2.** The canonical map $h : \overline{F_1} \to H/G_1$ is surjective and its restriction on $F_1$ is injective.

*Proof.* This action is surjective since for every $\tau \in H$, there exists $\gamma \in G_1$ such that $\gamma.\tau \in \overline{F_1}$ and injective since for a given $\tau_1$, $\tau_2 \in \overline{F_1}$ such that $\tau_2 = \gamma.\tau_1$, these elements are on the boundary of $\overline{F_1}$. □

**Theorem.** $\Gamma_1$ is generated by $S$ and $T$.

*Proof.* Let $\Gamma_1^{'}$ be the subgroup of $\Gamma_1$ generated by $S$ and $T$. Pick any point $\tau_0$ in the interior of $\overline{F_1}$, i.e., $F_1$. For any $\gamma \in \Gamma_1$, we must show that $\gamma \in \Gamma_1^{'}$. If $\tau = \gamma.\tau_0$, then there exists a $\gamma^{'} \in \Gamma_1^{'}$ such that $\gamma^{'}.\tau = \gamma^{'}\gamma.\tau_0 \in \overline{F_1}$. But since $\tau_0$ was chosen in $F_1$, and $\tau_0$, $\gamma^{'}\gamma.\tau_0$ are both in $\overline{F_1}$, so this implies that $\gamma^{'}\gamma = \pm I_2$ and hence $\gamma \in \Gamma_1^{'}$. □

## 5.3 Genus 2

$G_2 := \Gamma_2 / \langle \pm I_4 \rangle$ acts on the space $H_2$ via $\tau \to (A\tau + B)/(C\tau + D)^{-1}$ for all $\gamma = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in G_2$, where $A, B, C$ and $D$ are $2 \times 2$ integer matrices.

For genus 2, we have the following theorem due to Gottschling [63]:

**Theorem 7.** Under this action (5.7) a fundamental domain for $H_2/G_2$ is given by the subset of $\tau = \begin{pmatrix} \tau_1 & \tau_3 \\ \tau_3 & \tau_2 \end{pmatrix} \in H_2$ such that:

1. $|\mathcal{R}(\tau_j)| \leq \frac{1}{2}$ for all $j \in [1,3]$,
2. $\mathcal{I}(\tau)$ is simple Minkowski reduced.
3. $|\det(C\tau + D)| \geq 1$ for all symplectic matrices $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \{R_j\}_{1 \leq j \leq 19}$.

Gottschling [63] has shown that for the case $g = 2$, the necessary $C$ and $D$ in condition 3 are $C = I_2$ and $D$ is one of 15 choices, explicitly determined, all with entries $0, e = \pm 1$,

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} e & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & e \end{pmatrix}, \begin{pmatrix} e & 0 \\ 0 & e \end{pmatrix}, \begin{pmatrix} e & 0 \\ 0 & -e \end{pmatrix}, \begin{pmatrix} 0 & e \\ e & 0 \end{pmatrix}, \begin{pmatrix} e & e \\ e & 0 \end{pmatrix}, \begin{pmatrix} 0 & e \\ e & e \end{pmatrix}$$

or $C$ is a rank 1 matrix in which case, if $\tau = \begin{pmatrix} \tau_1 & \tau_3 \\ \tau_3 & \tau_2 \end{pmatrix}$ then $|\tau_1|, |\tau_2| \geq 1$ and $|\tau_1 + \tau_2 - 2\tau_3 \pm 1| \geq 1$.

From (5.2) or (5.4), we can now find the matrices $A$ and $B$ and construct the $\{R_j\}_{1 \leq j \leq 19}$ symplectic matrices below.

$$R_1 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \; R_2 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \; R_3 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$R_4 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \; R_5 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}, \; R_6 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$R_7 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}, \; R_8 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \; R_9 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$R_{10} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \; R_{11} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix}, \; R_{12} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$R_{13} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix},\ R_{14} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix},\ R_{15} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & -1 \end{pmatrix}$$

$$R_{16} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},\ R_{17} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},\ R_{18} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{pmatrix}$$

$$R_{19} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ -1 & 1 & 0 & -1 \end{pmatrix}.$$

Gottschling [63] has shown in his article the necessity and sufficiency of these 19 matrices to define Siegel's fundamental domain for genus 2. Subsequently, we will use in this context an algorithm written by Dupont [44] (see, algorithm 12).

where,

$$\mathcal{R}_1 = \begin{pmatrix} I_2 & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ 0 & I_2 \end{pmatrix},\ \mathcal{R}_2 = \begin{pmatrix} I_2 & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ 0 & I_2 \end{pmatrix},\ \mathcal{R}_3 = \begin{pmatrix} I_2 & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ 0 & I_2 \end{pmatrix}.$$

and $\{R_j\}_{1 \leq j \leq 19}$ are those introduced in theorem 7.

This algorithm follows from theorem 7, as well as from the following result:

**Lemma.** For all $\tau \in H_2$ and $\epsilon > 0$, the set

$$\left\{ \lambda \geq \epsilon \mid \exists \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in G_2 \text{ such that } \lambda = |\det(C\tau + D)| \right\}$$

is finite

**Proof.** It is a direct consequence of [[90], Lemma1, p.29].

**Algorithm 12** Reduce $H_2$ to $F_2$

**Input**:$\tau \in H_2$
**Output**:$(\gamma, \tau^{'}) \in G_2 \times F_2$ such that $\tau^{'} = \gamma.\tau$
$\gamma \leftarrow I_4$;
$\tau^{'} \leftarrow \tau$;
$t \leftarrow true$;
**while** $t$ **do**
  $U \leftarrow \text{MinkowskiReduction}(\mathcal{I}(\tau^{'}))$;
  $\gamma \leftarrow \begin{pmatrix} U^{\top} & 0 \\ 0 & U^{-1} \end{pmatrix} \gamma$;
  $\tau^{'} \leftarrow U^{\top}\tau^{'}U$;
  **for** $j = 1$ **to** 3 **do**
    $a \leftarrow -\left[\mathcal{R}(\tau_j^{'})\right]$;
    $\tau^{'} \leftarrow \mathcal{R}_j^a \tau^{'}$;
    $\gamma \leftarrow \mathcal{R}_j^a \gamma$;
  **end for**
  $t \leftarrow \text{false}$;
  **for** $j = 1$ **to** 19 **do**
    **if** $\left|\det(C_j\tau^{'} + D_j)\right| < 1$ **then**
      $t \leftarrow true$;
      $\tau^{'} \leftarrow R_j\tau^{'}$;
      $\gamma \leftarrow R_j\gamma$;
    **end if**
  **end for**
**end while**

## 5.4 Genus 3

In this section, we present some results concerning the fundamental domain for genus 3, in particular for rank $C = 1$. Before the discussion of our new results, we state a result due to Gottschling which plays a central role in our work.
We start by giving some definitions that are important for some details in this result.

**Definition.** Let $\mathcal{B}$ be a region containing $F_2$ which is defined by the following inequalities:

1. The standard limits for the real part of $\tau = \begin{pmatrix} \tau_1 & \tau_3 \\ \tau_3 & \tau_2 \end{pmatrix} = \begin{pmatrix} x_1 & x_3 \\ x_3 & x_2 \end{pmatrix} + i \begin{pmatrix} y_1 & y_3 \\ y_3 & y_2 \end{pmatrix} \in H^2$ :
$$-\frac{1}{2} \leq x_1, x_2, x_3 \leq \frac{1}{2};$$
2. The simple Minkowski conditions for $g = 2$:
$$y_2 \geq y_1 \geq 2y_3 \geq 0;$$
3. The two inequalities
$$|\tau_1| \geq 1, |\tau_2| \geq 1$$
that correspond to the third condition of $F_2$ and which are obtained by the following set of matrices:
$$C = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } C = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, D = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

**Definition.** Let $\mathcal{B}_0$ be the part of $\mathcal{B}$ in which $y_2 \leq 1$, $\mathcal{B}_1$ be that part of $\mathcal{B}_0$ in which $-\frac{1}{2} \leq x_3 \leq -\frac{1}{4}$ and $\mathcal{B}_{-1}$ that part of $\mathcal{B}_0$ in which $\frac{1}{4} \leq x_3 \leq \frac{1}{2}$.

**Lemma 4.** In the domain $\mathcal{B}_e$ where $(e = -1, 0, 1)$, the inequality
$$|\tau_1 + \tau_2 - 2\tau_3 - 2e| \geq 1;$$
is a sequence of $|\det(\tau + eS)| \geq 1$ with $S = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

*Proof.* See, [lemma 2, p. 109, [63]]. □

**Definition.** Two pairs $(C,\, D)$ and $(C_1,\, D_1)$ are associated when there is an unimodular matrix $U$ such that

$$(C_1 D_1) = U(CD).$$

### 5.4.1 $F_3$

It follows from Siegel's fundamental domain that its third condition in theorem 10 makes this domain not easy to be obtained, specially for $g \geq 3$. The task here, is to find a finite set (in other words, a finite number of conditions) for which the inequality, $|\det(C\tau + D)| \geq 1$, must be verified.

**Theorem.** For genus 3 and a rank 1 matrix $C$, we have the following inequalities to be verified:

$$\text{Writing } \tau = \begin{pmatrix} \tau_1 & \tau_4 & \tau_5 \\ \tau_4 & \tau_2 & \tau_6 \\ \tau_5 & \tau_6 & \tau_3 \end{pmatrix} \text{ then}$$

$$|\tau_1| \geq 1,\ |\tau_2| \geq 1,\ |\tau_3| \geq 1,\ |\tau_1 + \tau_2 - 2\tau_4 \pm 1| \geq 1, |\tau_2 + \tau_3 - 2\tau_6 \pm 1| \geq 1,$$

$$|\tau_1 + \tau_3 + 2\tau_5 \pm 1| \geq 1 \text{ if } \mathcal{I}(\tau_5) < 0,\ |\tau_1 + \tau_3 - 2\tau_5 \pm 1| \geq 1 \text{ if } \mathcal{I}(\tau_5) > 0,$$

$$|\tau_1 + \tau_2 + \tau_3 + 2\tau_4 - 2\tau_5 - 2\tau_6 + d| \geq 1,\ \text{for } -4 \leq d \leq 4,\ \text{and } \mathcal{I}(\tau_5) > 0,$$

$$|\tau 1 + \tau_2 + \tau_3 - 2\tau_4 - 2\tau_5 + 2\tau_6 + d| \geq 1, \text{for } -4 \leq d \leq 4,\ \text{and } \mathcal{I}(\tau_5) > 0,$$

$$|\tau_1 + \tau_2 + \tau_3 - 2\tau_4 + 2\tau_5 - 2\tau_6 + d| \geq 1,\ \text{for } -4 \leq d \leq 4,$$

*Proof.* Rank $C = 1$, therefore, there exist two unimodular matrices $U$ and $V$ such that

$$UCV = \begin{pmatrix} c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\ c > 0 \text{ and } UDV^{-\top} = \begin{pmatrix} d & d_3 & d_4 \\ d_5 & d_1 & d_6 \\ d_7 & d_8 & d_2 \end{pmatrix}.$$

It follows from (5.4), i.e., $CD^\top = DC^\top$ and $DA^\top - CB^\top = I_3$, that $d_5 = d_7 = 0$, $\gcd(c, d) = 1$ and $-d_6 d_8 + d_1 d_2 = 1$.

If we replace $U$ by $\begin{pmatrix} 1 & d_3 & d_4 \\ 0 & d_1 & d_6 \\ 0 & d_8 & d_2 \end{pmatrix} U$, then

$$UCV = \begin{pmatrix} c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ and } UDV^{-\top} = \begin{pmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.13}$$

Let

$$V^{-1} = \begin{pmatrix} p & q & r \\ s & t & u \\ v & w & x \end{pmatrix}, \text{ hence } V = \begin{pmatrix} tx - uw & rw - qx & qu - rt \\ uv - sx & px - rv & rs - pu \\ sw - tv & qv - pw & pt - sq \end{pmatrix},$$

with

$$p(tx - uw) - q(sx - uv) + r(sw - vt) = 1.$$

Finally, we obtain

$$|\det(C\tau + D)| = \left|c(p^2\tau_1 + q^2\tau_2 + r^2\tau_3 + 2pq\tau_4 + 2pr\tau_5 + 2qr\tau_6) + d\right|, \tag{5.14}$$

$$\text{with } \gcd(c, d) = 1 \text{ and } p\mathbb{Z} + q\mathbb{Z} + r\mathbb{Z} = 1. \tag{5.15}$$

And the matrices

$$C = \begin{pmatrix} c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^{-1}, \; D = \begin{pmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} V^{\top} \tag{5.16}$$

are coprime, symmetric (i.e., $CD^{\top} = DC^{\top}$ and $DA^{\top} - CB^{\top} = I_3$) and rank $C = 1$.

We will show now that the assignment between the classes of associated pairs $(C,\, D)$ with rank $C = 1$ and the integers $c, d, p, q, r$ is uniquely determined with the properties (5.15).
In the first hand, let, in addition of $V^{-1}$, $V_1^{-1}$ be an unimodular matrix with $(p, q, r)$, the first line of this matrix, and we take the following pair of matrices

$$C_1 = \begin{pmatrix} c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} V_1^{-1}, \; D_1 = \begin{pmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} V_1^{\top}, \tag{5.17}$$

similarly to (5.16).
It follows from (5.16), and (5.17) that

$$C_1 = U_1 C, \; D_1 = U_1 D,$$

where $U_1$ is an unimodular matrix of the form

$$U_1 = \begin{pmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} V_1^{\top} V^{-\top} \begin{pmatrix} d^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Hence, we can deduce that the pairs $(C_1,\, D_1)$ and $(C,\, D)$ are associated and by specifying $c, d, p, q, r$, the class of the pairs associated with $C$, $D$, is

uniquely determined. On the other hand, let $C$, $D$ be given (in the inverse way) and

$$U_0 C V_0 = \begin{pmatrix} c_0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; U_0 D V_0^{-\top} = \begin{pmatrix} d_0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where the first line of $V_0^{-1}$ is denoted by $(p_0, q_0, r_0)$ and we apply the corresponding conditions on $c, d, p, q, r$ to the integers $c_0, d_0, p_0, q_0, r_0$.
From

$$U_0 U^{-1} \begin{pmatrix} c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} c_0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} V_0^{-1} V,$$

and $c, c_0 > 0$, we deduce that $c = c_0$. We have also

$$V_0^{-1} V = \begin{pmatrix} \pm 1 & 0 & 0 \\ \star & \star & \star \\ \star & \star & \star \end{pmatrix}$$

which implies that the first lines of $V_0^{-1}$ and $V^{-1}$ coincide.
And from

$$U U_0^{-1} \begin{pmatrix} d_0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} d & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} V^{\top} V_0^{-\top}$$

we notice that $d_0 = d$.
According to (5.14), $|\det(C\tau + D)| \geq 1$ is equivalent to

$$\Big| c(p^2\tau_1 + q^2\tau_2 + r^2\tau_3 + 2pq\tau_4 + 2pr\tau_5 + 2qr\tau_6) + d \Big| \geq 1$$

where the integers $c, d, p, q, r$ satisfy the conditions (5.15).

The right-hand side of (5.14) can be estimated by the absolute value of the imaginary part, where the imaginary part is denoted by

$$\mathcal{I}(\tau) = \begin{pmatrix} y_1 & y_4 & y_5 \\ y_4 & y_2 & y_6 \\ y_5 & y_6 & y_3 \end{pmatrix}$$

therefore,

$$|\det(C\tau + D)| \geq c(p^2 y_1 + q^2 y_2 + r^2 y_3 + 2pq y_4 + 2pr y_5 + 2qr y_6).$$

Since the imaginary part of $\tau$ is Minkowski reduced, $\mathcal{I}(\tau)$ must satisfy the following Minkowski conditions:

$$y_1 \leq y_2 \leq y_3, \; 0 \leq y_4 \leq \frac{y_1}{2}, \; |y_5| \leq \frac{y_1}{2}, \text{ and } 0 \leq y_6 \leq \frac{y_2}{2}. \tag{5.18}$$

And from lemma 3, we obtain

$$y_1 \geq \frac{\sqrt{3}}{2}.$$

All these conditions allow us to write

$$\begin{aligned}
|\det(C\tau + D)| &= c(p^2y_1 + q^2y_2 + r^2y_3 + 2pqy_4 + 2pry_5 + 2qry_6) \\
&= \frac{c}{2}(p^2y_1 + q^2y_2 + 4pqy_4 + p^2y_1 + r^2y_3 + 4pry_5 + q^2y_2 + r^2y_3 + 4qry_6) \\
&\geq \frac{c}{2}y_1\left[(|p| - |q|)^2 + (|p| - |r|)^2 + (|q| - |r|^2\right] \\
&\geq \frac{c}{2}\frac{\sqrt{3}}{2}\left[(|p| - |q|)^2 + (|p| - |r|)^2 + (|q| - |r|^2\right] \\
&\geq \frac{\sqrt{3}}{4}\left[(|p| - |q|)^2 + (|p| - |r|)^2 + (|q| - |r|^2\right]
\end{aligned}$$

Clearly, this last inequality is greater or equal to 1 if and only if

$$(|p| - |q|)^2 + (|p| - |r|)^2 + (|q| - |r|^2) \geq 3,$$

and less than 1 when $|p|$, $|q|$ and $|r|$ are not too far from each other.
Hence, two cases must be considered:

1. $(|p| - |q|)^2 + (|p| - |r|)^2 + (|q| - |r|)^2 = 2$,
2. $(|p| - |q|)^2 + (|p| - |r|)^2 + (|q| - |r|)^2 = 0$.

It follows from 2, and $p \geq 0$ that $|p| = |q| = |r|$. Since $p\mathbb{Z} + q\mathbb{Z} + r\mathbb{Z} = 1$, $|p| = |q| = |r| = 1$ are the unique possibilities which imply four cases to be studied:

$$(p, q, r) = \{(1, 1, 1); (1, -1, 1); (1, 1, -1); (1, -1, -1)\}.$$

For $(p, q, r) = (1, 1, 1)$ :
we have,

$$|\det(C\tau + D)| \geq c\,(y_1 + y_2 + y_3 + 2y_4 + 2y_5 + 2y_6) > 1.$$

For $(p, q, r) = (1, -1, 1)$ :

$$|\det(C\tau + D)| \geq c\,(y_1 + y_2 + y_3 - 2y_4 + 2y_5 - 2y_6),$$

we notice from the fundamental domain of Minkowski, see [176], that the 16 systems of linear inequalities have the following condition:

$$y_1 + y_2 - 2y_4 + 2y_5 - 2y_6 \geq 0. \tag{5.19}$$

For $c \geq 2$, we notice that $|\det(C\tau + D)| > 1$, this remains the case where $c = 1$. Now, when $c = 1$, we obtain

$$|\det(C\tau + D)| = |\tau_1 + \tau_2 + \tau_3 - 2\tau_4 + 2\tau_5 - 2\tau_6 + d|$$

but since $\gcd(c, d) = 1$, we deduce that $|\det(C\tau + D)| \geq 1$ for $|d| \geq 5$. This leaves the following inequalities to be verified:

$$|\tau_1 + \tau_2 + \tau_3 - 2\tau_4 + 2\tau_5 - 2\tau_6 + d| \geq 1, \text{ for } -4 \leq d \leq 4. \tag{5.20}$$

For $(p, q, r) = (1, 1, -1)$:
we have,

$$|\det(C\tau + D)| \geq c\,(y_1 + y_2 + y_3 + 2y_4 - 2y_5 - 2y_6)\,.$$

From Minkowski's conditions (5.18), we notice that $|\det(C\tau + D)| > 1$ for $c \geq 2$ and as above it remains the true case where $c = 1$.
For $c = 1$ and $y_5 \leq 0$:

$$|\det(C\tau + D)| \geq y_1 + y_2 + y_3 + 2y_4 - 2y_5 - 2y_6 > 1.$$

However, if $c = 1$ and $y_5 > 0$, the following inequalities must be verified:

$$|\tau_1 + \tau_2 + \tau_3 + 2\tau_4 - 2\tau_5 - 2\tau_6 + d| \geq 1, \text{ for } -4 \leq d \leq 4 \text{ and } y_5 > 0. \tag{5.21}$$

For $(p, q, r) = (1, -1, -1)$:
we obtain,

$$|\det(C\tau + D)| \geq c\,(y_1 + y_2 + y_3 - 2y_4 - 2y_5 + 2y_6)\,.$$

As above, from Minkowski's conditions (5.18), we have $|\det(C\tau + D)| > 1$ for $c \geq 2$. It remains the true case where $c = 1$ which implies the following inequalities to be verified:

$$|\tau_1 + \tau_2 + \tau_3 - 2\tau_4 - 2\tau_5 + 2\tau_6 + d|\,, \text{ for } -4 \leq d \leq 4 \text{ and } y_5 > 0. \tag{5.22}$$

From 1, we find three possibilities:

- 1.*a*. $|p| - |q| = \pm 1$, $|p| - |r| = \pm 1$ and $|q| - |r| = 0$,
- 1.*b*. $|p| - |q| = \pm 1$, $|p| - |r| = 0$ and $|q| - |r| = \pm 1$,
- 1.*c*. $|p| - |q| = 0$, $|p| - |r| = \pm 1$ and $|q| - |r| = \pm 1$.

Since $p \geq 0$, for each possibility, three cases must be studied:

- 1.*d*. $p \geq 0$, $q \geq 0$ and $r \leq 0$,
- 1.*e*. $p \geq 0$, $q \leq 0$ and $r \geq 0$,
- 1.*f*. $p \geq 0$, $q \leq 0$ and $r \leq 0$.

We notice in 1.*d*, that if we replace $p$ by $p+1$, $q$ by $q+1$ and $r$ by $r-1$, and by Minkowski's conditions (5.18), then

$$\begin{aligned}|\det(C\tau + D)| &\geq c(p^2y_1 + q^2y_2 + r^2y_3 + 2pqy_4 + 2pry_5 + 2qry_6) \\ &= \frac{c}{2}[p^2y_1 + q^2y_2 + 4pqy_4 + p^2y_1 + r^2y_3 + 4pry_5 + q^2y_2 + r^2y_3 + 4qry_6] \\ &> 1.\end{aligned}$$

In 1.*f*, similarly, we replace $p$ by $p+1$, $q$ by $q-1$ and $r$ by $r-1$ and then by Minkowski's conditions (5.18), we deduce that $|\det(C\tau + D)| > 1$. Again, in 1.*e*, we replace $p$ by $p+1$, $q$ by $q-1$ and $r$ by $r+1$. Then by Minkowski's (5.18) and Tammela's (5.19) conditions, we obtain

$$\begin{aligned}|\det(C\tau + D)| &\geq c\left(p^2y_1 + q^2y_2 + r^2y_3 + 2pqy_4 + 2pry_5 + 2qry_6\right) \\ &> 1.\end{aligned}$$

Therefore, it remains to test the first values taken by $p$, $q$ and $r$.
We deduce from 1.*a*, in particular, when $|p| = 1 + |q|$ and $|q| = |r|$ that for $c = 1$ and $(p, q, r) = (1, 0, 0)$, the following inequality should be verified:

$$|\tau_1| \geq 1. \tag{5.23}$$

Again, from 1.*a*, when $|p| = -1 + |q|$ and $|q| = |r|$, we obtain the following conditions

$$|\tau_2 + \tau_3 - 2\tau_6 + d| \geq 1,\ (1, d) = 1$$

which correspond to $(p, q, r) = (0, 1, -1)$ and $c = 1$. And according to lemma 4, it is enough to check the following inequalities:

$$|\tau_2 + \tau_3 - 2\tau_6 \pm 1| \geq 1. \tag{5.24}$$

In 1.*b*, on the one hand, $|p| = 1 + |q|$ and $|p| = |r|$, implies the following conditions:

$$|\tau_1 + \tau_3 - 2\tau_5 + d| \geq 1,\ \text{(1,d)=1, and } y_5 > 0$$

which correspond to $c = 1$ with $(p, q, r) = (1, 0, -1)$.

$$|\tau_1 + \tau_3 + 2\tau_5 + d| \geq 1, \text{(1,d)=1, and } y_5 < 0$$

which correspond to $c = 1$ with $(p, q, r) = (1, 0, 1)$.
Again, according to lemma 4, it is enough to verify the following inequalities:

$$|\tau_1 + \tau_3 - 2\tau_5 \pm 1| \geq 1, \text{ for } y_5 > 0, \tag{5.25}$$

and

$$|\tau_1 + \tau_3 + 2\tau_5 \pm 1| \geq 1, \text{ for } y_5 < 0. \tag{5.26}$$

On the other hand, $|p| = -1+|q|$ and $|p| = |r|$, implies the following inequality:

$$|\tau_2| \geq 1, \tag{5.27}$$

and which corresponds to $c = 1$ with $(p, q, r) = (0, 1, 0)$.
In 1.$c$, $|p| = |q|$ and $|p| = 1 + |r|$ implies

$$|\tau_1 + \tau_2 - 2\tau_4 + d| \geq 1, \ (1, d) = 1$$

which correspond to $c = 1$ with $(p, q, r) = (1, -1, 0)$.
But according to lemma 4, we obtain the following inequalities:

$$|\tau_1 + \tau_2 - 2\tau_4 \pm 1| \geq 1. \tag{5.28}$$

Finally, $|p| = |q|$ and $|p| = -1 + |r|$ gives us the last inequality to be verified:

$$|\tau_3| \geq 1, \tag{5.29}$$

and which corresponds to $c = 1$ with $(p, q, r) = (0, 0, 1)$. This proves the theorem. □

## 5.5 Approximation to the Siegel fundamental domain

Whereas Siegel's fundamental domain as defined in Def 10 is an important theoretical concept in symplectic geometry, its practical relevance is limited since no constructive approach exists to actually identify the domain for $g > 2$: the first condition on the components of the matrix of the real part of $\Omega$ is straight forward. But as discussed in section 5.1, the Minkowski fundamental domain appearing in the second condition of Def 10 is only known for $g \leq 3$, and the third condition of Def 10 is, however, the least studied one.

For this reason, Siegel in [171] gave an algorithm to approximately reach the fundamental domain. Now, we review this algorithm due to Siegel which has been implemented together with the LLL algorithm in [38]. This algorithm is used here together with an exact determination of the shortest lattice vector (see, [51]).
Siegel proved the following:

**Theorem 8.** Any Riemann matrix $\Omega = X+iY \in H^g$ with real and imaginary part $X$ respectively $Y = R^\top R$, where $R$ is an upper triangular matrix, can be transformed by a symplectic transformation 5.7 to a matrix satisfying the following conditions:

1. $|X_{nm}| \leq 1/2$, for $n, m = 1, ..g$,
2. the squared length of the shortest lattice vector of the lattice generated by $Y$ is greater than or equal to $\sqrt{3}/2$.

The proof in [171], see also [38], is constructive and leads naturally to an algorithm:

*Proof.* The first condition can be always achieved by an appropriate choice of the matrix $B$ in $G_g = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$, $B = [X]$, i.e., each component of $B$ is the integer part of the corresponding component of $X$.
For the second condition, we assume that the shortest vector of the lattice generated by $R$, where $R$ is the Cholesky decomposition of $Y = R^\top R$, is the first vector of $R$. It is discussed in the previous chapters 2 and 4 that this can be always achieved. Siegel showed that the determinants of the imaginary part of two matrices $\tilde{\Omega} = \tilde{X} + i\tilde{Y}$ and $\Omega = X + iY$ related by a symplectic transformation (5.7) satisfy

$$\left|\det(\tilde{Y})\right| = \frac{|\det(Y)|}{|\det(C\Omega + D)|^2}. \tag{5.30}$$

If one considers the quasi-inversion (5.10), (5.11), equation (5.30) takes the form

$$\left|\det(\tilde{Y})\right| = \frac{|\det(Y)|}{|\Omega_{11}|^2}. \tag{5.31}$$

This leads to the following algorithm:

1. choose $A$ in (5.7) such that the shortest lattice vector appears as the first vector of $R$;
2. choose $B$ in (5.7) such that the real part of $\hat{\Omega} = A^\top \Omega A$ has components $\left|\hat{X}_{nm}\right| \leq 1/2$, for $n, m = 1, ..., g$;
3. if $\left|\hat{\Omega}_{11}\right| \geq 1$, terminate the algorithm; if not, apply the quasi-inversion (5.10), (5.11) and continue with step 1 of the algorithm for the resulting $\Omega$.

Because of (5.31) the modulus of the determinant of the imaginary part of the transformed matrix increases with each application of step 3. Since Siegel [171] has shown that there exists only a finite number of symplectic transformations leading to increasing $|\det(Y)|$ and that this determinant will be eventually greater than or equal to 1, the algorithm terminates after a finite number of steps. Then $Y_{11}$ is the squared length of the shortest lattice vector by construction. According to lemma 3, one has $Y_{11} \geq \frac{\sqrt{3}}{2}$. This proves the theorem. □

This algorithm can be used to efficiently compute multi-dimensional theta functions.

### 5.5.1 Theta functions

**Definition.** Theta functions with characteristics are defined as an infinite series. Let $\Omega$ be a $g \times g$ Riemann matrix, then

$$\Theta_{\mathrm{pq}}(\mathrm{z}, \Omega) = \sum_{\mathrm{N} \in \mathbb{Z}^g} \exp\left\{ i\pi \left\langle \Omega\left(\mathrm{N}+\mathrm{p}\right), \mathrm{N}+\mathrm{p} \right\rangle + 2\pi i \left\langle \mathrm{z}+\mathrm{q}, \mathrm{N}+\mathrm{p} \right\rangle \right\} , \quad (5.32)$$

with $\mathrm{z} \in \mathbb{C}^g$ and the *characteristics* $\mathrm{p}, \mathrm{q} \in \mathbb{R}^g$, where $\langle \cdot\cdot \rangle$ denotes the Euclidean scalar product $\langle \mathrm{N}, \mathrm{z} \rangle = \sum_{i=1}^{g} N_i z_i$. The positive definiteness of $\mathcal{I}(\Omega)$ guarantees the convergence of the series (5.32) for all values of z. Then the series (5.32) converges in both z and $\Omega$, and uniformly on compact sets. Therefore, the theta function with characteristics is an entire function of $\mathrm{z} \in \mathbb{C}^g$.

A characteristic is called *singular* if the corresponding theta function vanishes identically. Of special interest are half-integer characteristics with $2\mathrm{p}, 2\mathrm{q} \in \mathbb{Z}^g$. Such a half-integer characteristic is called *even* if $4\langle \mathrm{p}, \mathrm{q} \rangle = 0 \bmod 2$ and *odd* otherwise. It can be easily shown that theta functions with odd (even) characteristic are odd (even) functions of the argument z. The theta function with characteristic is related to the Riemann theta function $\Theta$, the theta function with zero characteristic $\Theta := \Theta_{00}$, via

$$\Theta_{\mathrm{pq}}(\mathrm{z}, \Omega) = \Theta(\mathrm{z} + \Omega\mathrm{p} + \mathrm{q}) \exp\left\{ i\pi \left\langle \Omega\mathrm{p}, \mathrm{p} \right\rangle + 2\pi i \left\langle \mathrm{p}, \mathrm{z}+\mathrm{q} \right\rangle \right\} . \quad (5.33)$$

From its definition, a theta function has the periodicity properties

$$\Theta_{\mathrm{pq}}(\mathrm{z}+\mathrm{e}_j) = \mathrm{e}^{2\pi i p_j} \Theta_{\mathrm{pq}}(\mathrm{z}) , \quad \Theta_{\mathrm{pq}}(\mathrm{z}+\Omega\mathrm{e}_j) = \mathrm{e}^{-2\pi i (z_j + q_j) - \mathrm{i}\pi\omega_{jj}} \Theta_{\mathrm{pq}}(\mathrm{z}) , \quad (5.34)$$

where $\mathrm{e}_j$ is a vector in $\mathbb{R}^g$ consisting of zeros except for a 1 in $j$th position. These periodicity properties ( 5.34) can be conveniently used in the computation of the theta function: an arbitrary vector $\mathrm{z} \in \mathbb{C}^g$ can be written in the

form $\mathrm{z} = \hat{\mathrm{z}} + \mathrm{N} + \Omega\mathrm{M}$ with $\mathrm{N}, \mathrm{M} \in \mathbb{Z}^g$, where $\hat{\mathrm{z}} = \Omega\hat{\mathrm{p}} + \hat{\mathrm{q}}$ with $|\hat{p}_i| \leq 1/2$, $|\hat{q}_i| \leq 1/2$. Thus, it is enough to compute the theta function for arguments $\hat{z}$ lying in the fundamental domain of the Jacobian, i.e., $\mathbb{C}^g/\Lambda$, where $\Lambda$ is the period lattice[1] formed by $\Omega$ and the $g$-dimensional identity matrix, $\hat{\mathrm{z}} = \Omega\hat{\mathrm{p}} + \hat{\mathrm{q}}$ with $|\hat{p}_i| \leq 1/2$, $|\hat{q}_i| \leq 1/2$. For general arguments z one computes $\Theta(\hat{z}, \Omega)$ and obtains $\Theta(\mathrm{z}, \Omega)$ from the periodicity properties (5.34) by multiplying with an appropriate exponential factor.

The convergence of the series (5.32) depends on the bilinear term, more precisely on the *shortest vector* $\mathrm{N}_{min}$ of the lattice $\mathbb{Z}^g$ equipped with the inner product defined by the imaginary part $Y$ of the Riemann matrix $\Omega$: $\langle \mathrm{N}, \mathrm{M} \rangle_Y := \langle Y\mathrm{N}, \mathrm{M} \rangle$, $\mathrm{N}, \mathrm{M} \in \mathbb{Z}^g$ (see, [22, 38, 51]). For a given Riemann matrix the shortest vector $\mathrm{N}_{min}$ is then defined in terms of its squared length

$$y_{min} = \langle \mathrm{N}_{min}, \mathrm{N}_{min} \rangle_Y := \min_{\mathrm{N} \in \mathbb{Z}^g/\{0\}} \langle Y\mathrm{N}, \mathrm{N} \rangle . \tag{5.35}$$

The longer the shortest vector, the more rapid the convergence of the theta series. This idea follows from the fact that the theta series can be approximated by a sum, $|N_i| \leq N_\epsilon$, $i = 1, .., g$, where the constant $N_\epsilon$ is chosen such that all omitted terms in (5.32) are smaller than some prescribed value of $\epsilon$ where $\epsilon = 10^{-16}$. We sum over a $g$-dimensional sphere and we take into account that we can choose z in the fundamental domain of the Jacobian because of (5.34), we get with (5.35) for the Riemann theta function the estimate

$$N_\epsilon > \sqrt{-\frac{\ln \epsilon}{\pi y_{min}}} + \frac{1}{2},$$

(see, [51, 22]). Changing the shortest vector can be achieved by changing the homology basis of the underlying Riemann surface which yields a different but symplectically equivalent Riemann matrix. This can be achieved by using modular transformations, i.e., symplectic transformations with integer coefficients to generate larger norms of the shortest vector in order to accelerate the convergence of a theta series for given $\Omega$. This part corresponds to the step 3 of the algorithm presented in the proof of theorem 8. Since the behavior of theta functions under modular transformations is explicitly known, such transformations can dramatically increase the rate of convergence which is especially important for larger values of $g$. This approach was for the first time implemented in an algorithm by Deconinck et. al. in [38].

The main task in this context is the identification of the shortest vector in a given $g$-dimensional lattice known as the shortest vector problem (SVP).

[1]Note, that this lattice $\Lambda$ is not to be confused with the lattice generated by the matrix $Y$ discussed in this work.

Currently, there is no known algorithm that would solve this problem in polynomial-time. The LLL algorithm yields an approximation to the shortest vector in polynomial-time but with an error growing exponentially with the dimension $g$ (though in practice slowly with $g$ such that it can be used for small genus as an approximation). For this reason in, [38] the SVP was solved approximately via the LLL algorithm. However, since we are interest in an evaluation of theta functions in a large number of points, it can be beneficial to identify the shortest vector exactly even for small $g$. Though it is computationally demanding this knowledge will accelerate the ensuing evaluation of the theta function (5.35). Therefore, we replace the LLL algorithm in [38] with an exact Minkowski reduction for $g \leq 5$, and with an exact solution to the SVP for higher genus.

### 5.5.2 Example

As an example we want to study the Riemann matrix of the Fricke-Macbeath surface [56, 115], a surface of genus $g = 7$ with the maximal number $84(g-1) = 504$ of automorphisms. It can be defined via the algebraic curve

$$f(x, y) := 1 + 7yx + 21y^2x^2 + 35x^3y^3 + 28x^4y^4 + 2x^7 + 2y^7 = 0. \quad (5.36)$$

The code [52] produces for this curve the following Riemann matrix[2]

```
RieMat =

  Columns 1 through 4

   1.0409 + 1.3005i   0.0530 + 0.3624i   0.3484 + 0.0000i
   0.0530 + 0.3624i  -0.5636 + 1.0753i   0.0187 - 0.5975i
   0.3484 + 0.0000i   0.0187 - 0.5975i   1.0544 + 1.7911i
   0.2077 + 0.6759i   0.6749 + 0.3001i   0.3220 - 1.0297i
  -0.2091 - 0.2873i   0.1220 - 0.5274i   0.3029 + 0.8379i
  -0.1064 - 0.4257i   0.1205 - 0.1783i  -0.2297 - 0.3668i
   0.3590 + 0.5023i   0.1990 - 0.1118i   0.3495 - 0.0499i

  Columns 5 through 7

   0.2077 + 0.6759i  -0.2091 - 0.2873i  -0.1064 - 0.4257i
   0.6749 + 0.3001i   0.1220 - 0.5274i   0.1205 - 0.1783i
```

[2]For the ease of the reader, we present only 4 digits though the Riemann matrix is computed with an error of the order of $10^{-10}$.

```
  0.3220 - 1.0297i   0.3029 + 0.8379i  -0.2297 - 0.3668i
 -0.0978 + 1.7041i  -0.7329 - 0.8055i  -0.0714 - 0.1766i
 -0.7329 - 0.8055i   1.1824 + 1.0163i   0.4425 + 0.2592i
 -0.0714 - 0.1766i   0.4425 + 0.2592i   0.2815 + 0.7791i
 -0.0415 + 0.5448i   0.0835 - 0.2430i  -0.6316 - 0.0369i

Columns 7 through 7

   0.3590 + 0.5023i
   0.1990 - 0.1118i
   0.3495 - 0.0499i
  -0.0415 + 0.5448i
   0.0835 - 0.2430i
  -0.6316 - 0.0369i
   0.2315 + 0.6895i.
```

**Remark.** Since we work with finite precision, rounding is an issue also in the context of lattice reductions. The code [52] generally produces results with a tolerance Tol between $10^{-10}$ and $10^{-14}$, which appears for instance in the form of an asymmetry of the computed Riemann matrix of the order of Tol. Since in lattice reductions the components of the Riemann matrix are multiplied with integers, these errors will be amplified. Thus a rounding of an order of magnitude larger than Tol is necessary in practice.

After LLL reduction the first basis vector of the lattice is found to have squared norm 1.3005 i.e., the first component of the imaginary part of the above Riemann matrix. Note that the lattice basis is almost LLL reduced, there are only minor effects of the LLL algorithm applied to this matrix. Since the norm of the shortest vector is greater than $\sqrt{3}/2$, no quasi-inversion is applied. An ensuing shift of the real part leads to the matrix

```
W =

  Columns 1 through 3

   0.0409 + 1.3005i   0.0530 + 0.3624i  -0.4849 - 0.6245i
   0.0530 + 0.3624i   0.4364 + 1.0753i  -0.3594 - 0.6598i
  -0.4849 - 0.6245i  -0.3594 - 0.6598i  -0.4706 + 1.3844i
```

```
 -0.1064 - 0.4257i   0.1205 - 0.1783i  -0.1946 - 0.1178i
  0.3590 + 0.5023i   0.1990 - 0.1118i  -0.0510 - 0.0073i
 -0.4511 + 0.1383i  -0.0171 + 0.2485i  -0.0543 - 0.3239i
  0.2684 - 0.2975i  -0.4161 + 0.2521i   0.0481 + 0.3949i

 Columns 4 through 6

 -0.1064 - 0.4257i   0.3590 + 0.5023i  -0.4511 + 0.1383i
  0.1205 - 0.1783i   0.1990 - 0.1118i  -0.0171 + 0.2485i
 -0.1946 - 0.1178i  -0.0510 - 0.0073i  -0.0543 - 0.3239i
  0.2815 + 0.7791i   0.3684 - 0.0369i   0.3907 - 0.1531i
  0.3684 - 0.0369i   0.2315 + 0.6895i   0.3656 - 0.1563i
  0.3907 - 0.1531i   0.3656 - 0.1563i  -0.4318 + 0.6585i
 -0.2437 - 0.3094i  -0.2134 - 0.1308i  -0.1541 + 0.0260i

Columns 7 through 7

 0.2684 - 0.2975i
-0.4161 + 0.2521i
 0.0481 + 0.3949i
-0.2437 - 0.3094i
-0.2134 - 0.1308i
-0.1541 + 0.0260i
-0.4997 + 1.0021i.
```

However, the square of the norm of the shortest lattice vector of the imaginary part of the matrix $W$ is 0.6585, well below the threshold $\sqrt{3}/2$. Note that the LLL reduced $\tilde{Y}$ above has the shortest vector in the 6th column (with squared norm 0.6585). One could construct a unimodular matrix $Z$ such that $RZ$ has this vector appearing in the first column (the resulting matrix might not satisfy the LLL condition ). This would be more suited to the application of Siegel's algorithm, but will be still approximate since in general LLL does not identify the shortest lattice vector correctly.

If the same algorithm is applied with an exact determination of the shortest vector, the picture changes considerably: in the first step of the iteration, the shortest lattice vector is correctly identified having the square of the norm 0.6585. Thus after a shift of the real part, a quasi-inversion is applied. The subsequent identification of the shortest vector of the resulting matrix leads to a vector of squared norm 0.7259. After a shift of the real part, another

quasi-inversion is applied. This time the square of the norm of the shortest vector is 1.0211 and thus greater than $\sqrt{3}/2$. After a shift of the real part we finally obtain $W=$

```
Columns 1 through 3

 0.3967 + 1.0211i   0.0615 - 0.1322i  -0.0000 + 0.0000i
 0.0615 - 0.1322i   0.3967 + 1.0211i   0.3553 - 0.5828i
-0.0000 + 0.0000i   0.3553 - 0.5828i   0.2894 + 1.1656i
-0.4609 - 0.2609i  -0.3386 + 0.1933i   0.0905 + 0.2450i
 0.3553 - 0.5828i   0.4776 - 0.1287i  -0.4776 + 0.1287i
 0.1838 + 0.3219i   0.2743 + 0.5669i   0.3871 - 0.3736i
-0.3386 + 0.1933i  -0.3386 + 0.1933i  -0.1223 - 0.4541i


Columns 4 through 6

-0.4609 - 0.2609i   0.3553 - 0.5828i   0.1838 + 0.3219i
-0.3386 + 0.1933i   0.4776 - 0.1287i   0.2743 + 0.5669i
 0.0905 + 0.2450i  -0.4776 + 0.1287i   0.3871 - 0.3736i
 0.3967 + 1.0211i  -0.4776 + 0.1287i   0.0167 - 0.3895i
-0.4776 + 0.1287i   0.2894 + 1.1656i  -0.1671 - 0.7115i
 0.0167 - 0.3895i  -0.1671 - 0.7115i   0.4414 + 1.2784i
 0.0615 - 0.1322i   0.0905 + 0.2450i  -0.3386 + 0.1933i

Columns 7 through 7

-0.3386 + 0.1933i
-0.3386 + 0.1933i
-0.1223 - 0.4541i
 0.0615 - 0.1322i
 0.0905 + 0.2450i
-0.3386 + 0.1933i
 0.3967 + 1.0211i.
```

For applications, it is important to know how long a certain task takes on a given computer. For the above example, the LLL code is not very efficient, but converges in roughly 1 $ms$. The SVP code takes in this case $4-5$ times

longer, which is still completely negligible compared to what can be gained by applying the above algorithm.

If we consider an example of even higher genus, the curve

$$f(x, y) := y^9 + 2x^2y^6 + 2x^4y^3 + x^6 + y^2 = 0 \tag{5.37}$$

of genus 16, we find a similar behavior. Using Siegel's algorithm on the Riemann matrix for this curve computed with the code [52], we find that the variant with the LLL algorithm converges within three iterations. The LLL algorithm takes $1 - 2ms$ in each step. The algorithm produces $\Omega_{11} = 0.3314 + 1.0188i$, a value clearly larger than 1. The length of the shortest vector generated by the imaginary part of this Riemann matrix as found via SVP is 0.4437, well below the theoretical minimum of $\sqrt{3}/2 \approx 0.866$. On the other hand Siegel's algorithm with an exact solution of the SVP in each step requires 14 iterations where each SVP takes around $10ms$. Finally we get $\Omega_{11} = 0.4748 + 0.8956i$, i.e., a shortest vector almost twice as long as what has been found with the LLL algorithm. As we see, the approximative LLL algorithm is for $g < 20$ only an order of magnitude faster than the SVP algorithm, but finds the shortest vector merely with an error growing exponentially with $g$.

# Bibliography


[1] N. A'Campo, L. Ji, A. Papadopoulos, *On the early history of moduli and Teichmüller spaces*, L. Keen, I. Kra and R. E. Rodriguez. Lipman Bers, a Life in Mathematics, American Mathematical Society, 978-1-4704-2056-7, pp. 175-262, 2015.

[2] L. Afflerbach, *Minkowskische Reduktionsbedingungen für positiv definite quadratische Formen in 5 Variablen*, Mh. Math. 94, pp. 1-8, 1982.

[3] L. Afflerbach and H. Grothe, *Calculation of Minkowski-reduced lattice bases*, computing, vol. 35, no. 3-4, pp. 269-276, 1985.

[4] M. Agrawal, N. Kayal and N. Saxena, *PRIMES is in P*, Ann. of Math. (2) 160, pp. 781-793, 2004.

[5] E. Agrell, T. Eriksson, A. Vardy and K. Zeger, *Closest point search in lattices*, IEEE Trans. Inform. Theory, vol. 48, no. 8, pp. 2201-2214, Aug. 2002.

[6] D. Aharonov and O. Regev, *Lattice problems in NP and co-NP*, J. ACM, 52, pp. 749-765, Sept 2005.

[7] M. Ajtai, *The shortest vector problem in $L_2$ is NP-hard for randomized reductions*, in Proc. 30-th Annual ACM Symp. Theory of Computing, pp. 193-203, Dallas, TX, May 1998.

[8] M. Ajtai and C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence*, In Proc of 29th STOC, pp. 284-293. ACM, 1997.

[9] M. Ajtai, R. Kumar, and D. Sivakumar, *A Sieve algorithms for the shortest lattice vector problem*, in Proc. ACM STOC'01, Crete, Greece, pp. 601-610, Jul. 2001.

[10] M. Ajtai, R. Kumar, and D. Sivakumar, *Sampling short lattice vectors and the closest lattice vector problem*, In Proc. of 17th IEEE Annual Conference on Computational Complexity (CCC), pp. 53-57, 2002.

[11] A. Akhavi, *The optimal LLL algorithm is still polynomial in fixed dimension*, Theor. Comput. Sci., vol. 297, no.1, pp. 3-23, Mars. 2003.

[12] J. D. Alper, *Oracle Theory*, Oracle Theory courses Notes, May 2001.

[13] A. N. Andrianov, *Quadratic forms and Hecke operators*, Grundlehren Math. Wiss. 286. Springer, Berlin, 1987.

[14] S. Aurora, B. Barak, *Computational complexity: A modern approach*, Cambridge University Press, 2009.

[15] L. Babai, *On Lovász Lattice reduction and the nearest lattice point problem*, Combinatorica, 6, pp. 1-13, 1986.

[16] S. Bai, T. Laarhoven and D. Stehlé, *Tuple Lattice Sieving*, volume 19, Issue A(Algorithmic Number Theory symposium XII), LMS Journal of Computation and Mathematics, pp. 146-162, Jan 2016.

[17] A. H. Banihashemi and A. K. Khandani, *On the complexity of decoding lattices using the Korkine-Zolotarev reduced basis*, IEEE Trans. Inf. Theory. Vol. 44, no. 1, pp. 162-171, Jan. 1998.

[18] E. S. Barnes and M. J. Cohn, *On Minkowski reduction of positive quaternary quadratic forms*, Mathematika 23, pp. 156-158, Dec 1976.

[19] A. Becker, L. Ducas, N. Gama, and T. Laarhoven, *New directions in nearest neighbor searching with applications to lattice sieving*, In Proc. Of SODA, pp. 10-24. SIAM, 2016.

[20] W. A. Beyer, R. B. Roof and D. Williamson, *The lattice structure of multiplicative congruential pseudo-random vectors*, Math. Comput, 25, pp 345-360, 1971.

[21] H. F. Blichfeldt, *A new principle in the geometry of numbers, with some applications*, Transactions of the American Mathematical Society, vol. 15, pp. 227-235, 1914.

[22] A. I. Bobenko, C. Klein, *Computational Approach To Riemann Surfaces*, Lectures Notes in Mathematics 2013, Mathematics ISSN 0075-8434, Springer, pp. 152-155, 2013.

[23] H. W. Braden, T. P. Northover, *Klein's curve*, J. Phys. A 43 (2010), no. 43, 434009, 17 pp. 2010.

[24] M. R. Bremner, *Lattice basis reduction, An introduction to the LLL and its applications*, CRC Press, pp. 1-336, 2012.

[25] V. Brun, *En generalisation av kjederbrøken I*, Skv.Vidensk. Selsk. Kristiana, Mat. Nat. Klasse, vol. 6. pp. 1-29, 1919.

[26] V. Brun, *En generalisation av kjderbrøken II*, Skv.Vidensk. Selsk. Kristiana, Mat. Nat. Klasse, Vol. 6. pp. 1-24, 1920.

[27] J.-Y. Cai and T. W. Cusick, *A lattice-based Public-key cryptosystem, Information and Computation*, vol. 151. Issues 1-2, pp. 17-31, 1999.

[28] J. W. S. Cassels, *An introduction to the geometry of numbers*, Springer-Verlag, Berlin, 1971.

[29] X. W. Chang and G. H. Golub, *Solving ellipsoïd-constrained integer least squares problems*, SIAM J. Matrix Anal. Appl., vol. 31, no. 3, pp. 1071-1089, 2009.

[30] H. CHEN and L. XU, *Greedy Algorithm Computing Minkowski Reduced Lattice Bases with Quadratic Bit Complexity of Input Vectors*, Chin. Ann. Math. Ser. B 32:857, doi: 10.1007/s11401-011-0680-1, pp. 857-862, 2011.

[31] I. V. L. Clarkson, *Approximation of linear forms by lattice points with applications to signal processing*, PhD. dissertation, Australian Nat. Univ., Canberra, Australia, 1997.

[32] O. Coldreich, S. Goldwasser and S. Halevi, *Public-key cryptosystem from lattice Reduction Problems*, Advances in cryptology-CRYPTO, vol. 1294 of LNCS, pp. 112-131, Springer-Verlag, 1997.

[33] J. H. Conway, N. J. A Sloane, *Sphere Packings Lattices and Groups*, Third Edition, Grundlehrender mathematschen Wissenschaften, 3. Band. vol 290, Springer-Verlag, 1999.

[34] J. H. Conway and N. J. A. Sloane, *On the Voronoi regions of certain lattices*, SIAM Journal on Algebraic and Discrete Methods, vol. 5, pp. 294-305, Sept. 1984.

[35] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, New York, NY: Springer-Verlag, 3rd ed., 1999.

[36] D. Coppersmith, *Small solutions to polynomial equations, and low exponent RSA vulnerabilities*, J. of Cryptology, 10(4), pp. 233-260, 1997.

[37] H. Daudée and B. Vallée, *An upper bound on the average number of iterations of the LLL algorithm*, Theor. Comput. Sci., vol. 123, no. 1, p. 95-115, Jan. 1994.

[38] B. Deconinck, M. Heil, A. Bobenko, M. van Hoeij, M. Schmies, *Computing Riemann theta functions*, Mathematics of Computation, 73, pp. 1417-1442, 2004.

[39] B. Deconinck, M. Hoeij, *Computing the Riemann matrix of algebraic curve*, PhysicaD, 152, pp. 28-46, 2001.

[40] J.L De Lagrange, *Recherches d'arithmétique*, Nouveaux Mémoires de l'Académie de Berlin, 1773.

[41] B. N. Delone, R. V. Galiulin, and M. I. Shtogrin, *On types of Bravais lattices*, in: Sovrem. Probl. Mat., Vol. 2, Moscow, pp. 119-254, 1973.

[42] E. De Shalit and E. Z. Goren, *On special values of theta functions of genus two*, Ann. Int. Fourier (Grenoble), 47(3), pp. 775-799, 1997.

[43] K. Draziotis and D. Poulakis, *Lattice attacks on DSA schemes based on Lagrange's algorithms*, 5th international conference on algebraic Informatics, CAI 2013, Berlin. LNCS 8080, pp. 119-131, 2013.

[44] R. Dupont, *Moyenne Arithmético-Géométrique, suites de Borchardt et applications*, thèse, 2006.

[45] F. Eisenbrand, *Integer Programming And Algorithmic Geometry Of Numbers*, A tutorial, Chapter of 50 Years of Integer Programming 1958-2008, pp. 505-559, Nov 2009.

[46] F. Eisenbrand, and G. Rote, *Fast reduction of ternary quadratic forms*, In Proceedings Of The 2001 Cryptography And lattices Conference ($CALC'01$). Lecture Notes in Computer Science, vol. 2146, Springer-Verlag, pp. 32-44, 2001.

[47] M. Euchner, *Praktische Algorithmen Zur Gitterreduktion Und Faktorisierung*, Diplomarbeit Uni. Frankfurt. 1991.

[48] J. D. Fay, *Theta functions on Riemann surfaces*, Lect. Notes in Math., 352, Springer, 1973.

[49] U. Fincke and M. Pohst, *Improved methods for calculating vectors of short length in a lattice, including a complexity analysis*, Math. Comput., vol. 44, no. 170, pp. 463-471, 1985.

[50] R. Fitzpatrick, C. Bischof, J. Buchamann, Ö. Dagdelen, F. Göpfert, A. Mariano, and B.-Y. Yang, *Tuning GaussSieve for speed*, In Proc. Of LATINCRYPT, volume 9230 of LNCS, pp. 288-305, Springer, 2015.

[51] J. Frauendiener, C. Jaber, C. Klein, *Efficient computation of multidimensional theta functions*, arXiv:1701.07486, 2017.

[52] J. Frauendiener, C. Klein, *Computational approach to compact Riemann surfaces*, Nonlinearity 30(1), 138, 2016.

[53] J. Frauendiener, C. Klein, *Computational Approach to Hyperelliptic Riemann Surfaces*, Lett. Math. Phy. 105(3), pp. 379-400, doi:10.1007/s11005-015-0743-4, 2015.

[54] J. Frauendiener, C. Klein, *Hyperelliptic theta functions and spectral methods: Kdv and KP solutions*, Lett. Math. Phy., 76, pp. 249-267, 2006.

[55] E. Freitag, *Siegelsche Modulfunktionen*, Grundlehren der Mathematischen wissenschaften 254, Springer-Verlag, Berlin, 1983.

[56] R. Fricke, *Über eine einfache Gruppe von* 504 *Operationen*, Mathematische Annalen, 52 (23), pp. 321-339, 1899.

[57] Y. H. Gan, G. Ling, and W. H. Mow, *Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection*, IEEE Trans. Signal Processing, vol. 57, no. 7, pp. 2701-2710, July 2009.

[58] Y. H. Gan and H. W. Mow, *Complex lattice reduction algorithms for low-complexity MIMO detection*, in Proc. IEEE Global Communications Conf. (GLOBECOM), St. Louis, MI, pp. 2953-2957, Nov 2005.

[59] M. R. Garey and D. S. Johnson, *Computers and intractability. A guide to the theory of NP-completeness*, A series of books in the Mathematical Sciences. W. H. Freeman and Co., San Francisco, Calif., 1979.

[60] C. F. Gauss, *Disquisitiones Arithmetics*, Springer-Verlag, 1801.

[61] C. F. Gauss, *Untersuchungen über höhere Arithmetik*, (Disquisi-tiones Arithmeticae). Berlin, Germany: Springer-Verlag, 1889.

[62] O. Goldreich, D. Micciancio, S. Safra, and J. -P. Seifert, *Approximating shortest lattice vectors is not harder than approximating closest lattice vectors*, Information Processing Letters, Vol. 71, pp. 55-61, July 1999.

[63] E. Gottschling, *Explizite Bestimmung der Randfächen des Fundamentalbereiches der Modulgruppe zweiten Grades*, Math. Ann. 138, pp. 103-124, 1959.

[64] R. L. Graham, M. Grötschel and L. Lovász, eds, *Handbook of combinatories*, Vol. 1,2. Elsevier, Amsterdam, 1995.

[65] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algotithms and combinatorial Optimization*, Berlin, Germany: Springer-Verlag, 1993.

[66] P. Gruber and C. Lekkerkerker, *Geometry of numbers*, North-Holland Publishing Co., 1987.

[67] J. Hadamard, *Résolution d'une question relative aux détérminants*, Bulletin des sciences math. (2), 17, pp. 240-248, 1893.

[68] R. Hain, *Lectures on moduli spaces of elliptic curves*, arXiv:0812.1803v3 [math.A6], Mars 2014.

[69] W. Hämäläinen, *Class NP, NP-Complete, and NP-Hard Problems*, Nov 2006.

[70] G. Hanrot and D. Stehlé, *Improved analysis of Kannan's shortest lattice vector algorithms (extended abstract)*, In Proceedings of Crypto 2007, Volume 4622 of Lecture Notes in Computer Science, pp. 170-186, Springer-Verlag, 2007.

[71] G. Hanrot and D. Stehlé, *Worst-case Hermite-Korkine-Zolotarev reduced lattice bases*, [Research Report] RR-6422, INRIA., pp. 25, <inria-00211875v2>, 2008.

[72] G. Hanrot, X. Pujol and D. Stehé, *Algorithms for the shortest and closest lattice vectors problems*, In IWCC, volume 6639 of LNCS, pp. 159-190, Springer, 2011.

[73] B. Hassibi and H. Vikalo, *On the sphere-decoding algorithm I: Expected complexity*, IEEE Trans. Signal Process., vol. 53, no. 8, pp. 2806-2818, Jul. 2005.

[74] B. Helfrich, *Algorithms to construct Minkowski reduced and Hermite reduced lattice bases*, Theory comput. Sci., vol. 41, no. 2-3, pp. 125-139, 1985.

[75] M. Henk, *Note on shortest and nearest lattice vectors*, Information Processing Letters, vol. 61, pp. 183-188, 1997.

[76] C. Hermite, *Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres*, J. Reine. Angew. Math., vol. 40, pp. 279-290, 1850.

[77] J. Hoffstein, J. C. Pipher and J. H. Silverman, *An introduction to mathematical cryptography*, Undergraduate texts in mathematics. Springer, 2008.

[78] F. Holland, *Another Proof Of Hadamard's Determinantal Inequality*, Irish Math. Soc. Bulletin 59, pp. 61-64, 2007.

[79] L. K. Hua, I. Reiner, *On the Generators of the Symplectic Modular group*, Transactions of the American Mathematical Society, vol. 65, no. 3, pp. 415-426, May 1949.

[80] J. Jaldén and B. Ottersen, *On the complexity of sphere decoding in digital communications*, IEEE Trans. Signal Process., vol. 53, no. 4, pp. 1474-1484, Mar. 2005.

[81] J. Jaldén, D. Seethaler, and G. Matz, *Worst-and average case complexity of LLL lattice reduction in MINO wireless systems*, in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, NV, pp. 2685-2688, Apr. 2008.

[82] L. Ji, J. Jost, *Universal moduli spaces of Riemann surfaces*, arXiv: 1611.08732v1 [math.A6], Nov. 2016.

[83] A. Joux and J. Stern, *Lattice reduction: A toolbox for the cryptanalyst*, J. Cryptol., vol. 11, no. 3, pp. 161-185, 1998.

[84] R. Kannan, *Improved algorithms for integer programming and related lattice problems*, In Proceedings of the 15th Symposium on the theory of computing (STOC 1983), pp. 99-108. ACM Press, 1983.

[85] R. Kannan, *Algorithmic geometry of numbers*, Annual review of computer science, 2, pp. 231-267, 1987.

[86] R. Kannan, *Minkowski's convex body theorem and integer programming*, Math. Oper. Res., vol. 12, pp. 415-440, Aug. 1987.

[87] R. Kannan, *Improved algorithms for integer programming and related lattice problems*, In Proceedings of the fifteenth annual ACM symposium on Theory of Computing, STOC' 83, pp. 99-108, NEW YORK, NY, USA, ACM, 1983.

[88] R. Kannan, A. K. Lenstra and L. Lovász, *Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers*, in, Proc. 16th Ann. ACM Symp. on Theory of computing, Washington, D.C., pp. 191-200, 1984.

[89] E. Kaplan, *LLL Algorithm*, Lattices in computer Science, Tel Aviv University, Fall 2004.

[90] H. Klingen, *Introductory Lectures on Siegel Modular Forms*, Cambridge: Cambridge University Press, 1990.

[91] D. E. Knuth, *The Art of Computer Programming*, Reading, 2nd ed Reading, MA: Addison-Wesley, 1981.

[92] A. Korkine and G. Zolotareff, *Sur les formes quadratiques*, Math. Ann., vol. 6, pp. 366-389, 1873.

[93] D. A. Korotkin, *Finite-gap solutions of the stationary axially symmetric Einstein equation in vacuo*, (Russian); translated from Teoret. Mat. Fiz. 77(1988), no. 1, pp. 25–41 Theoret. and Math. Phys. 77(1988), no. 1, pp. 1018-1031, 1989.

[94] A. Krieg, *Primitive minima of positive definite quadratic forms*, Acta Arithmetica, 63:1, pp. 91-96, 1993.

[95] T. Laarhoven, *Sieving for shortest vectors in lattices using angular locality-sensitive hashing*, In Proc. Of CRYPTO, volume 9215 of LNCS, pp. 3-22. Springer, 2015.

[96] T. Laarhoven, M. Mosca, and J. van de Pol, *Finding shortest lattice vectors faster using quantum search*, Designs, Codes and Cryptography, 77(2-3), pp. 375-400, 2015.

[97] J. C. Lagarias, *Worst-Case complexity bounds for algorithms in the theory of integral quadratic forms* J. Algorithms 1, pp. 142-186, 1980.

[98] 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 (4), pp. 333-348, Springer-Verlag, 1990.

[99] A. Lamacchia, *Basis Reduction Algorithms And Subset Sum Problems*, SM Thesis, Dept. of Elect. Eng. Comp. Sci, Massachusetts Institute of Technology, Cambridge, MA 1991.

[100] C. G. Lekkerkerker, *Geometry of Numbers*, Wolters-Noordhoff, Groningen, 1969.

[101] A. K. Lenstra, *Lattices and factorization of polynomials*, ACM, vol. 15 Issue 3, August 1981.

[102] H. W. Lenstra, Jr., *Integer programming with a fixed number of variables*, Math. Oper. Res.8, pp. 538-548, 1983.

[103] A. K. Lenstra, H. Lenstra Jr and L. Lovász, *Factoring Polynomials With Rational Coefficients*, Math. Ann., 261, pp. 515-534, Dec 1982.

[104] L. Likavec, *Application of lattice basis reduction*, thesis, Technische Universität Darmstadt, mars 2011.

[105] C. Ling, *Towards Characterizing the performance of Approximate Lattice decoding in MIMO communications*, Proc. Int. Symp. Turbo codes and ITG conf. Source channel coding, 2006.

[106] C. Ling and N. Howgrave-Graham, *Effective LLL reduction for lattice decoding*, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Nice, France, Jun. 2007.

[107] C. Ling and N. Howgrave-Graham, *Effective LLL reduction for lattice decoding*, in Proc. IEEE Int. Symp. Information Theory (ISIT), Nice, France, pp. 196-200, 2007.

[108] L. Lovász, *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50. SIAM, CBMS-NSF Regional Conference Series in Applied Mathematics, 1986.

[109] L. Lovász and H. Scarf, *The generalized basis reduction algorithms*, Math. Oper. Res., vol 17, Issue 3, pp. 751-764, 1992.

[110] F. T. Luk and D. M. Tracy, *An improved LLL algorithm*, Linear Algebra Appl., vol 428, no. 2-7, pp. 441-452, Jan. 2008.

[111] F. T. Luk and S. Qiao, *A pivoted LLL algorithm*, Linear Algebra., vol. 434, no. 11, pp. 2296-2307, Jun. 2011.

[112] F. T. Luk, S. Qiao and W. Zhang, *A lattice basis reduction algorithm*, nstitute for computational Mathematics, Hong Kong, Baptist University, Tech. Rep. 10-04, Apr. 2010.

[113] X. Ma, W. Zhang, and A. Swami, *Lattice-reduction aided equalization for OFDM systems*, IEEE Trans. Wireless Commun., vol. 8, no. 4, pp. 1608-1613, Apr. 2009.

[114] R. Macausland, *The Moore-Penrose Inverse and Least Squares*, Math 420: Advanced Topics in linear algebra, university of Puget Sound, Creative commons License, April 2014.

[115] A. Macbeath, *On a curve of genus* 7, Proceedings of the London Mathematical Society 15, pp. 527-542, 1965.

[116] K. Mahler, *On Minkowski's theory of reduction of positive quadratic forms*, Quart. J. Math. 9, pp. 259-262, 1938.

[117] K. Mahler, *On reduced positive definite ternary quadratic forms*, J. London Math. Soc. 15, pp. 193-195, July 1940.

[118] K. Mahler, *On reduced positive definite quaternary quadratic forms*, Nieuw Arch. Wiskunde (2) 22, pp. 207-212, May 1946.

[119] K. Mahler, *A theorem on inhomogeneous diophantine inequalities*, Nederl. Akad. Wetensch., Proc. 41, pp. 634-637, 1938.

[120] G. Marsaglia, *The structure of linear congruential sequences*, Applications Of Number Theory to Numerical Analysis ($S.K.Zaremba$), ed, pp. 249-285, 1972.

[121] H. Minkowski, *Über die positiven quadratischen Formen und über kettenbruchähnliche Algorithmen*, J. Reine und Angewandte Math., vol. 107, pp. 278-297, 1891.

[122] H. Minkowski, *Gesammelte Abhandlungen 1*, pp. 145-148, 153-156, 217-218. Leipzig-Berlin: Teubner 1911.

[123] H. Minkowski, *Diskontinuitatsbereich für arithmetische Äquivalenz.*, Ges. Abh., vol. 2, Leipzig-Berlin, pp. 53-100, 1911.

[124] D. Micciancio, *The LLL Algorithm*, CSE 206A: Lattice Algorithms and Applications, UCSD CSE, Winter 2012.

[125] D. Micciancio, *The shortest vector in a lattice is NP-hard to approximate to within some constant*, in Proc. 39-th Annual Symp. Found. Computer Science, pp. 92-98, Palo Alto, CA, Nov, 1998.

[126] D. Micciancio, *Lecture 3: Minimum distance*, CSE 206A: Lattice Algorithms and Applications, Spring 2007.

[127] D. Micciancio, *Introduction To Lattices*, CSE 206A: Lattice Algorithms And Applications, UCSD CSE, Winter 2010.

[128] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A cryptographic Perspective*, Boston, MA: Kluwer Academic, 2002.

[129] D. Micciancio an P. Voulgaris, *Faster exponential time algorithms for the shortest vector problems*, in Proc. ACM/SIAM SODA '01, Austin, TX, pp. 1468-1480, Jan 2010.

[130] D. Micciancio and P. Voulgaris, *A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations*, In Proc. STOC'10, pp. 351-358. ACM, 2010.

[131] I. Morel, D. Stehlé, and G. Villard, *H-LLL: Using householder inside LLL*, in Proc. Int. Symp. on Symb and Alg. Comput. (ISSAC' 09), Seoul, Korea, pp. 271-278, Jul. 2009.

[132] H. W. Mow, *Universal lattice decoding: A review and some recent results*, in Proc. IEEE Int. Conf. Communications (ICC), Paris, France, vol. 5, pp. 2842-2846, 2004.

[133] D. Mumford, *Tata lectures on Theta. I and II.*, Progress in Mathematics, 28 and 43, respectively. Birkhäuser Boston, Inc., Boston, MA, 1983 and 1984.

[134] C. E. Nelson, *The reduction of positive definite quinary quadratic forms*, Aequationes Math. 11, pp. 163-168, 1974.

[135] A. Neumaier and D. Stehlé, *Faster LLL-type reduction of lattice bases*, ISSAC' 16, Waterloo, Ontorio, Canada. ACM., DOI: $10.475/123_4$, 2016.

[136] P. Q. Nguyen and D. Stehlé, *An LLL algorithm with quadratic complexity*, SIAM J. Comput ., vol. 39, no. 3, pp. 874-903, 2009.

[137] P. Q. Nguyen and D. Stehlé, *Low dimensional basis reduction revisited*, In. D. A. Buell, editor, Proceedings of ANTS 2004, number 3076 in LNCS, pp. 338-357, Springer-Verlag, 2004, ACM Transactions on Algorithms, 5(4), 2009.

[138] P. Nguyen and J. Stern, *Lattice Reduction in cryptology: An Update*, W. Bosma (Ed.): ANTS-IV, LNCS 1838, pp. 85-112, Springer-Verlag Berlin Heidelberg, 2000.

[139] P. Q. Nguyen and B. Vallée, *The LLL Algorithm: Survey and Applications*, Eds. Berlin, Germany: Springer-Verlag, 2009.

[140] P. Q. Nguyen and T. Vidick, *Sieve algorithms for the shortest vector problem are practical* J. Math. Crypt., vol. 2, no. 2, pp. 181-207, 2008.

[141] A. M. Odlyzko, *The rise and fall of Knapsack cryptosystems*, In Cryptology and Computational Number Theory, volume 42 of Proc. of Symposia in Applied Mathematics, pp. 75-88. A.M.S., 1990.

[142] M. Pohst, *On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications*, ACM SIGSAM Bulletin, vol. 15, pp. 37-44, Feb. 1981.

[143] D. Poulakis, *New lattice attacks on DSA Schemes*, J. Mathematical Cryptololy, vol. 10, Issue 2, pp. 135-144, 2016.

[144] X. Pujol and D. Stehlé, *Rigorous and efficient short lattice vectors enumeration*, In Proc. ASIACRYPT'08, vol. 5350 of LNCS, pp. 390-405, Springer, 2008.

[145] X. Pujol and D. Stehlé, *Solving the shortest lattice vector problem in time* $2^{2.465n}$, Cryptology ePrint Archive, Report 2009/605, pp. 1-7, 2009.

[146] S. Qiao, *A Jacobi method for lattice basis reduction*, In Proceeding of 2012 International Conference On Wireless Communications and Networks, Xi'an China, May 2012.

[147] S. Radziszowski and D. Kreher, *Solving Subset Problems With The LLL algorithm*, J. Combin. Math. Combin. Comput. 3, pp. 48-63, 1988.

[148] O. Regev, *Lecture notes of lattices in Computer Science*, taught at the Computer Science Tel Aviv university, Fall 2009.

[149] B. Riemann, *Theorie der Abel'schen Functionen*, Journal für die reine und angewandte Mathematik, pp. 115-155, 1857.

[150] S. S. Ryshkov, *On the reduction of positive quadratic forms of n variables in the sense of Hermite, Minkowski, and Venkov*, Dokl. AN SSSR, 207, no. 5, pp. 1054-1056, 1972.

[151] S. S. Ryshkov, *On the theory of reduction of positive quadratic forms*, Dokl. AN SSSR, 198, no. 5, pp. 1028-1031, 1971.

[152] S. S. Ryshkov, *On the reduction theory of positive quadratic form*, J. Soviet Math. Dokl. 12, pp. 946-950, 1971.

[153] S. S. Ryshkov, *The theory of Hermite-Minkowski reduction of positive definite quadratic forms*, J. Soviet Math. 6, pp. 651-671, 1976.

[154] S. S. Ryshkov, *On the theory of Hermite-Minkowski reduction of positive quadratic forms*, Zap. Nauchn. Sem. Leningr. Otd. Mat. Inst., vol. 33, Leningrad, pp. 37-64, 1973.

[155] C. Saha, P. Verna, *Lecture* 14*: Randomized / Probabilistic computation*, E 224 computational complexity theory, Aug-Dec 2015.

[156] C. P. Schnorr, *A more efficient algorithms for lattice basis reduction*, J. Algorithms 9, pp. 47-62, 1988.

[157] C. P. Schnorr, *A hierarchy of polynomial lattice basis reduction algorithms*, Theor. Comput. Sci., vol. 53, no. 2-3, pp. 201-224, 1987.

[158] C. P. Schnorr, *Average time fast SVP and CVP algorithms for low density lattices*, TR Goethe Universität Frankfurt, Jan 2010.

[159] C. P. Schnorr and M. Euchner, *Lattice basis reduction: Improved practical algorithms and solving subset sum problems* Mathematical Programming, vol. 66, pp. 181-191, 1994.

[160] C. P. Schnorr and H. H. Hörner, *Attacking The Chor-Rivest Cryptosystem By Improved Lattice Reduction*, In Proc. EUROCRYPT'95, LNCS 921, Springer-Verlag, Berlin New York, pp. 1-12, 1995.

[161] D. Seethaler and G. Matz, *Efficient vector perturbation in multi-antenna multi-user systems based on approximate integer relations*, in Proc. European Signal Processing Conf. (EUSIPCO), Florence, Italy, Sept, 2006.

[162] D. Seethaler, G. Matz, and F. Hlawatsch, *Low-complexity MIMO detection using Seysen's lattice reduction algorithm*, in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, HI, pp. 53-56, Apr. 2007.

[163] Y. Sheng, *Relationships between Nondeterministic and Deterministic tape complexity*, Math. 336, 2014.

[164] I. Semaev, *A* 3*-dimensional lattice reduction algorithm*, In Proceedings Of The 2001 Cryptography And Lattices Conference ($CALC'01$). Lecture Notes in Computer Science. Vol. 2146. Springer-Verlag, pp. 181-193, 2001.

[165] M. Seysen, *Simultaneous reduction of a lattice basis and its reciprocal basis*, Combinatorica, vol. 13, Issue 3, pp. 363-376, Sept 1993.

[166] G. Shomonin, *Minkowski's theorem and its application*, Integer Points in polyhedra, 2009.

[167] C. L. Siegel, *Lectures on the Geometry of Numbers*, Springer-Verlag, 1989.

[168] C. L. Siegel, *Einführung in die Theorie der Modulfunktionen n-ten Grades*, Math. Ann. 116, pp. 617-657, 1939.

[169] C. L. Siegel, *Lectures on Quadratic Forms*, Notes by K. G. Ramanathan, Bombay: Tata institute of Fundamental Research, 1967.

[170] C. L. Siegel, *Symplectic Geometry*, Academic Press, New York and London, 1964.

[171] C. L. Siegel, *Topics in complex function theory*, vol. III. John Wiley and sons, Inc., New York, 1989.

[172] D. Simon, *Selected Applications Of LLL In Number Theory*, Chapter of The LLL algorithm, part of the series Information Security and Cryptography, pp. 265-282, 2009.

[173] N. J. A. Sloane, *The Sphere Packing Problem*, arXiv: math/020725v1 [math. CO], Jul 2012.

[174] D. Stehlé and M. Watkins, *On the extremality of an* 80*-dimensional lattice*, In: Hanrot G., Morain F., Thomé E. (eds) Algorithmic Number Theory. ANTS 2010. Lectures Notes in Computer Science, vol 6197, Springer, Berlin, Heidelberg, pp. 340-356, 2010.

[175] C. Swierczewski, B. Deconinck, *Computing Riemann theta functions in Sage with applications*, Mathematics and computers in Simulation 127, pp. 263-272, 2016.

[176] P. P. Tammela, *On reduction theory of positive quadratic forms*, Studies in number theory. Part 3, Zap. Nauchn. Sem. LOMI, 50, "Nauka". Leningrad. Otdel., Leningrad, pp. 6-96, 1975.

[177] P. P. Tammela, *The Hermite-Minkowski domain of reduction of positive definite quadratic forms in six variables*, Zap. Nauchen. Sem. Leningrad. Otdeh. Mat. Inst. Steklov. LOMI, 13, pp. 72-89; English transl. in J. Soviet. Math. 5, no. 3, 1976.

[178] P. P. Tammela, *Minkowski reduction region for positive quadratic forms in seven variables*, J. Sov. Math, vol. 16, Issue 1, pp. 836-857, 1981.

[179] P.P. Tammela, *Theory of reduction of positive-definite quadratic forms: Nonnormality of the partition of the positivity cone into Minkowski ($n \geq 7$) and Barnes-Cohn ($n = 4$) reduction region*, J. Sov. Math, vol. 43, Issue 5, pp. 2699-2705, Dec 1988.

[180] P. P. Tammela, *Reduction theory of positive quadratic forms*, J. Math Sci, vol. 11, Issue 2, pp. 197-277, 1979.

[181] Z. Tian and S. Qiao, *A Complexity Analysis Of a Jacobi Method for Lattice Basis Reduction*, Proceeding C3S2E'12 Proceedings of the Fifth International C* conference on Computer Science and Software Engineering, pp. 53-60, 2012.

[182] B. Vallée, *Gauss's algorithm revisited*, J. of Algorithm, 12(4), pp. 556-572, 1991.

[183] B. Vallée, *La réduction des réseaux: autour de l'algorithme de Lenstra, Lenstra, Lovász*, RAIRO Inform. Théor. Appl., 23(3), pp. 345-376, 1989. English translation in CWI Quaterly, 3(2), pp. 95-120, 1990.

[184] B. Vallée, *Une approche géométrique de la réduction des réseaux en petite dimension*, Ph.D. thesis, Université de Caen, 1986.

[185] B. L. Van der Waerden, *Die Reduktionstheorie der positiven quadratischen Formen*, Acta Math. 96, pp. 265-309, 1956.

[186] B. L. Van der Waerden, *Das Minimum von $D/f_{11}f_{22}...f_{55}$ für reduzierte positive quinäre quadratische Formen*, Aequationes Math. 2, pp. 233-247, 1969.

[187] B. L. Van Der Waerden and H. Gross, *Studien Zur Theorie der Quadratischen Formen*, Book, Mathematische Reihe, vol. 34, 1968.

[188] P. Van Emde Boas *Another NP-Complete partition problem and the complexity of computing short vectors in a lattice* Rep. 81-04, Mathematisch Institut, Amsterdam, The Netherlands, Apr. 1981.

[189] A. Vardy and Y. Be'ery, *Maximum-likelihood decoding of the Leech Lattice*, IEEE Trans. Inform. Theory, vol. 39, pp. 1435-1444, July 1993.

[190] A. Vera, *Analyses de l'algorithme de Gauss. Applications à l'analyse de l'algorithme LLL*, Algorithme et structure de données [cs.DS], thèse, Université de Caen, HAL, 2009.

[191] E. Viterbo and J. Boutros, *A universal lattice code decoder for fading channels*, IEEE Trans. Inf. Theory, vol. 45, no. 5, pp. 1639-1642, Jul. 1999.

[192] X. Wang, M. Liu, C. Tian and J. Bi, *Improved Nguyen-Vidick heuristic sieve algorithms for shortest vector problem*, Cryptology ePrint Archive, Report 2010/647, 2010.

[193] D. Wübben, R. Böhnke, V. Kühn and K. D. Kammeyer, *Near-maximum-likehood detection of MIMO systems using MMSE-based lattice reduction*, in Proc. Int. Commun. Conf. (ICC' 04), pp. 798-802, Jun. 2004.

[194] D. Wübben, R. Böhnke, V. Kühn, and K. D. Kammeyer, *MMSE-based lattice reduction for near-ML detection of MIMO systems*, in Proc. Int. ITG Workshop on Smart Antennas, Munich, Germany, pp. 106-113, Mar. 2004.

[195] D. Wübben, D. Seethaler, J. Jaldén, and G. Marz, *Lattice reduction: A survey with applications in wireless communications*, IEEE Signal Process. Mag., vol. 28, no. 3, pp. 70-91, May 2011.

[196] D. Zagier, *Elliptic modular forms and their applications*, In the 1-2-3 of Modular forms: Lectures at a summer school in Nordfjordeid, Norway (=[10] of "books":ed.K. Ranested), Universitext, Springer-verlag, Berlin-Heidelberg-New York, pp. 1-103, 2008.

[197] F. Zhao and S. Qiao, *Radius Selection Algorithms For Sphere Decoding*, C3S2E'09 Proceedings of the 2nd Canadian conference on Computer Science and Software Engineering, pp. 169-174, 2009.

[198] W. Zhang, S. Qiao and Y. Wei, *HKZ and Minkowski Reduction Algorithms for Lattice-Reduction-Aided MIMO Detection*, IEEE Transactions on SIGNAL Processing, vol. 60, no. 11, Nov. 2012.

[199] W. Zhang, S. Qiao and Y. Wei, *Practical HKZ and Minkowski Lattice Reduction Algorithms*, Dept. Comput. Software, McMaster University, Hamilton, ON, Canada, Tech. Rep. CAS-11-04-SQ, 2011.