Considerations on Computational Lattice Problems

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Urs Wagner

von

Untersiggenthal AG

Promotionskomitee

Prof. Dr. Joachim Rosenthal (Vorsitz, Leitung der Dissertation)
Dr. Gérard Maze
Prof. Dr. Chris Monico (Begutachter)
Prof. Dr. Josep Climent (Begutachter)

Zürich 2013

To my parents.

Acknowledgements

First I would like to thank my advisor Joachim Rosenthal. His support and extraordinary commitment created opportunities that would have stayed out of reach otherwise.

I am most deeply indebted to Gérard Maze who initiated this dissertation and kept on adding fertilizer throughout. I will —and already do— miss the regular discussions and brainstormings. They did not only lead to nice results, but were always also a great source of joy and inspiration.

Further I would like to thank all the other members of the workgroup —namely my academic sister Anna-Lena and Felix who knows 'everything about computers'— and the whole Institute of Mathematics of the University of Zurich for the excellent work environment.

Last but not least I would like mention my friends, my parents, my brothers and sister and my girlfriend Katrin. Thanks for all the love and support.

Abstract

Lattices are discrete subgroups of the Euclidean space. While they are highly structured objects and their elements can easily be described by means of integer linear combinations of their basis vectors, it is possible to define NP-hard problems on them. Due to the existence of a class of lattices with favourable worst-case to average-case connection, these problems are well suited as basis for provable secure cryptosystems. The problems appearing in this thesis are the shortest vector problem (SVP), the closest vector problem (CVP) and their approximate versions respectively. All known algorithms that solve these problems exactly run in exponential time, while there are polynomial time algorithms that solve these problems approximately. The latter ones either perform or are based on basis reduction. Any improvement on these algorithms has direct impact on the security parameters of various cryptographic functions proposed. This thesis contains results on the algorithmic side as well as theoretical considerations. Concerning approximation of SVP, a polynomial time improvement of LLL —probably the most famous lattice basis reduction algorithm— is presented. Further, improvements on both, an algorithm to solve the SVP exactly and an algorithm to solve the CVP exactly, are given. The first extends the AKS sieve algorithm while the second is a closest point search algorithm based on HKZ bases. The more theoretical results are two-fold: on the one hand, we prove new inequalities on the harmonic, geometric and arithmetic means inequalities if two of the means are known. This leads to new inequalities between the orthogonality defect and the Seysen measure of a lattice basis, both quantifying the reducedness of a basis. On the other hand, the natural density of rectangular unimodular matrices is computed, giving partial answers on the probability that randomly chosen lattice vectors can be completed into a basis or generate the lattice respectively.

Zusammenfassung

Gitter sind diskrete Untergruppen des euklidischen Vektorraumes, deren Elemente einfach als ganzzahlige Linearkombinationen ihrer Basisvektoren beschrieben werden können. Trotzdem bilden sie die Grundlage für verschiedene NP-schwere Probleme. Es existiert sogar eine Klasse von Gittern mit günstigem worst-case zu average-case Verhalten, was sie für die Entwicklung von beweisbar sicheren Kryptosystemen interessant macht.

In dieser Arbeit spielen die zwei wohl bekanntesten Gitterprobleme, das shortest vector problem (SVP) und das closest vector problem (CVP) bzw. die entsprechenden Näherungsprobleme, eine zentrale Rolle. Während Algorithmen, welche die exakten Versionen dieser Probleme lösen, exponentielle Laufzeiten haben, existieren Algorithmen mit polynomieller Laufzeit zur Lösung der approximativen Probleme. Die Letzteren basieren meist auf Gitterreduktionsalgorithmen und jede Verbesserung dieser hat einen direkten Einfluss auf die Sicherheitsparameter verschiedenster kryptographischer Funktionen.

Diese Arbeit beinhaltet sowohl algorithmische als auch theoretische Resultate. Im Bereich des approximativen Lösens von SVP wird eine Verbesserung des wohl berühmtesten Gitterreduktionsalgorithmus LLL präsentiert. Es wird bewiesen, dass der neue Algorithmus polynomielle Laufzeit hat. Weiter werden Verbesserungen von Algorithmen, welche die exakten Versionen von SVP und CVP lösen, vorgestellt. Die Eine basiert auf dem AKS sieve Algorithmus, die Andere auf einem CVP-Suchalgorithmus, welcher auf Basen operiert, deren dualen Basen HKZ reduziert sind.

Auf der eher theoretischen Seite beweisen wir einerseits neue Ungleichungen zwischen dem harmonischen, geometrischen und arithmetischen Mittel, falls zwei der drei Mittel gegeben sind. Dies führt zu neuen Ungleichungen zwischen dem orthogonality defect und der Seysen measure, zweier Werte, welche die Reduziertheit der Basis quantifizieren. Andererseits berechnen wir die asymptotische Dichte von invertierbaren Rechteckmatrizen. Dieses Resultat gibt eine Teilantwort auf die Frage, mit welcher Wahrscheinlichkeit zufällig gewählte Gittervektoren zu einer Basis ergänzt werden können, beziehungsweise mit welcher Wahrscheinlichkeit sie das Gitter erzeugen.

Contents

Chapter 1

Introduction

In this chapter we provide the reader with some necessary background on lattices and introduce the notation used throughout the thesis (Section 1.1). It is further explained how the work conduced is motivated (Section 1.2) and an overview on the results is given (Section 1.3).

1.1 Lattices

We consider lattices as discrete subgroups of $\mathbb{R}^m$ and unless stated otherwise, $\|\cdot\|$ denotes the Euclidean norm. For $1 \leq p < \infty$, $\|\cdot\|_p$ denotes the usual p-norm and $\|\cdot\|_\infty$ denotes the infinity norm. Bold lowercase letters (e.g. $\mathbf{b}$, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{v}$) will usually denote vectors. Bold uppercase letters (e.g. $\mathbf{S}$, $\mathbf{C}$) will usually denote sets.

Definition 1.1.1 *Let $n \leq m \in \mathbb{N}$. A* lattice $\mathcal{L}$ *of* dimension m *and* rank n *is a discrete subgroup of $\mathbb{R}^m$ consisting of all $\mathbb{Z}$-linear combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$. The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ form a* basis *of the lattice.*

If the rank n equals the dimension m we say that $\mathcal{L}$ has *full rank*. Given a lattice basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of $\mathcal{L}$ we will usually write them as columns of a matrix $B \in \mathbb{R}^{m \times n}$ in the following way:

$$B = [\mathbf{b}_1, \ldots, \mathbf{b}_n].$$

For a lattice $\mathcal{L}$, a subgroup $\mathcal{L}' \subseteq \mathcal{L}$ is called *sublattice* of $\mathcal{L}$. The elements of a lattice $\mathcal{L}$ with basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ are the integer linear combinations of the basis vectors,

$$\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \mathcal{L}(B) = \{B\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n\}.$$

It is important to notice that lattices of rank $n \geq 2$ have infinitely many bases, as the following equivalence implies:

$$\mathcal{L}(B_1) = \mathcal{L}(B_2) \Leftrightarrow \exists U \in \mathrm{GL}_n(\mathbb{Z}) : B_1 U = B_2.$$

It is clear that if two bases B_1 and B_2 generate the same lattice, they also span the same n-dimensional subspace in $\mathbb{R}^m$:

$$\mathcal{L}(B_1) = \mathcal{L}(B_2) \Rightarrow \operatorname{span}(B_1) = \operatorname{span}(B_2).$$

Hence for a lattice $\mathcal{L} \subset \mathbb{R}^m$, $\operatorname{span}(\mathcal{L})$ is well-defined as the space spanned by the vectors of any of its bases. Associated with a basis B of a lattice is its *fundamental parallelotope*

$$\mathcal{P}(B) := \big\{B\mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_\infty < 1\big\}.$$

The *volume* $\operatorname{vol}(\mathcal{L})$ of a lattice $\mathcal{L}$ of rank n and with basis B is the n-dimensional volume of its fundamental parallelotope and can be computed by

$$\operatorname{vol}\big(\mathcal{L}(B)\big) = \det\big(\mathcal{L}(B)\big) = \sqrt{\det B^T B} = \sqrt{\det(\langle \mathbf{b}_i, \mathbf{b}_j\rangle)_{i,j}}.$$

Note again that if $B_2 = B_1 U$ with $U \in \mathrm{GL}_n(\mathbb{Z})$, then $\det B_1^T B_1 = \det B_2^T B_2$ and hence the volume of a lattice is well-defined. For a given basis $B \in \mathbb{R}^{m\times n}$, we denote by π_i, $1 \le i \le n$, the orthogonal projection

$$\pi_i : \operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \longrightarrow \operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^\perp.$$

The usual Gram-Schmidt orthogonalized basis is then given by

$$B^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*] = [\pi_1(\mathbf{b}_1), \ldots, \pi_n(\mathbf{b}_n)].$$

Defining the Gram-Schmidt coefficients as

$$\mu_{i,j} := \frac{\langle \mathbf{b}_i, \mathbf{b}_j^*\rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^*\rangle} \quad \text{for} \quad 1 \le j < i \le n,$$

we have

$$\pi(\mathbf{b}_i) = \mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^* \tag{1.1.1}$$

and

$$B^T = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ \mu_{21} & 1 & \ddots & & \vdots \\ \mu_{31} & \mu_{32} & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \mu_{n1} & \cdots & \cdots & \mu_{nn-1} & 1 \end{pmatrix} B^{*T}. \tag{1.1.2}$$

The volume of a lattice can alternatively be computed from the lengths of the Gram-Schmidt basis vectors:

$$\operatorname{vol}\big(\mathcal{L}(B)\big) = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|.$$

For a positive real number $r > 0$ and a point $\mathbf{x} \in \mathbb{R}^m$, by $\mathcal{B}_m(\mathbf{x}, r) \subset \mathbb{R}^m$ we denote the closed ball of radius r around the origin:

$$\mathcal{B}_m(\mathbf{x}, r) := \{\mathbf{y} \in \mathbb{R}^m : \|\mathbf{y}\| \leq r\}.$$

In the case where $\mathbf{x} = \mathbf{0} \in \mathbb{R}^m$ we simply write $\mathcal{B}_m(r) := \mathcal{B}_m(\mathbf{0}, r)$. For $0 \leq r < R$ we denote the half open annulus by

$$\mathcal{C}_m(r, R) := \mathcal{B}_m(R) \backslash \mathcal{B}_m(r) = \{\mathbf{y} \in \mathbb{R}^m : r < \|\mathbf{y}\| \leq R\}.$$

We have seen that a lattice $\mathcal{L} \subset \mathbb{R}^m$ consists of points in the m-dimensional Euclidean space that are arranged in a highly structured way. Given a basis of a lattice, the elements of the group are easy to describe, and checking whether a point $\mathbf{t} \in \mathbb{R}^m$ belongs to the lattice can be done in polynomial time. Despite the easy description and seemingly nice structure of a lattice, there are some associated values which are not in general easy to compute. Some of them are the successive minimas and the covering radius which we are going to define now.

Definition 1.1.2 *The* i-th minimum $\lambda_i(\mathcal{L})$ *of a lattice* $\mathcal{L}$ *is defined as the radius of the smallest zero-centered ball containing at least* i *linearly independent lattice vectors:*

$$\lambda_i(\mathcal{L}) := \min\left\{r > 0 : \dim\left(\operatorname{span}\left(\mathcal{L} \cap \mathcal{B}_m(r)\right)\right) \geq i\right\}.$$

In particular $\lambda_1(\mathcal{L})$ denotes the length of the shortest nonzero lattice vector in $\mathcal{L}$.

Definition 1.1.3 *The* covering radius $\mu(\mathcal{L})$ *is defined as the maximum distance a point in* $\operatorname{span}(\mathcal{L}) \subseteq \mathbb{R}^m$ *can have from the lattice:*

$$\mu(\mathcal{L}) := \max_{\mathbf{t} \in \operatorname{span}(\mathcal{L})} \min_{\mathbf{v} \in \mathcal{L}} \|\mathbf{t} - \mathbf{v}\|.$$

Equivalently this is the smallest radius that closed balls centered at each lattice point must have to cover the whole space spanned by the lattice vectors.

As already mentioned, the computation of the successive minimas and the covering radius is not in general easy. However there are some well known bounds (e.g. [CSB87, MG02, NV10]): Recall that given a basis B of a lattice $\mathcal{L}$ in $\mathbb{R}^m$ its volume $\operatorname{vol}(\mathcal{L})$ can be computed in polynomial time. In fact the volume is a meaningful invariant of the lattice. The first minimum λ_1 of a lattice of rank n with given volume can be upper bounded. Let Λ_n denote the set of all lattices of rank n.

Definition 1.1.4 *The* Hermite constant *of dimension* n *is defined as*

$$\gamma_n := \max_{\mathcal{L} \in \Lambda_n} \frac{\lambda_1^2(\mathcal{L})}{\operatorname{vol}(\mathcal{L})^{2/n}},$$

where the maximum is taken over all lattices of rank n*.*

The values of γ_n along with the corresponding lattices are known for $n = 1, \dots, 8$ and $n = 24$. For the rest only bounds are known. Using Blichfeldts Theorem (see e.g. [MG02]), which says that in any measurable subset $\mathbf{S} \subset \text{span}(\mathcal{L})$ of volume bigger than $\text{vol}(\mathcal{L})$ there exist two distinct points $\mathbf{t}_1, \mathbf{t}_2 \in \mathbf{S}$ such that $\mathbf{t}_1 - \mathbf{t}_2 \in \mathcal{L}$, Minkowski was able to derive an upper bound on the Hermite constants. Setting

$$\mathbf{S} = \mathcal{B}_m(r) \cap \text{span}(\mathcal{L}) \quad \text{with} \quad r = \left(\frac{\text{vol}(\mathcal{L}) + \epsilon}{\text{vol}(\mathcal{B}_n(1))} \right)^{1/n},$$

we get that

$$\text{vol}(\mathbf{S}) = \text{vol}(\mathcal{L}) + \epsilon > \text{vol}(\mathcal{L}).$$

We can conclude what is basically Minkowski's first theorem:

$$\lambda_1(\mathcal{L}) \leq 2 \cdot \frac{\text{vol}(\mathcal{L})^{1/n}}{\text{vol}(\mathcal{B}_n(1))^{1/n}}. \tag{1.1.3}$$

Hence, for all lattices of rank n it holds that

$$\gamma_n \leq \frac{\lambda_1^2(\mathcal{L})}{\text{vol}(\mathcal{L})^{2/n}} \leq \frac{4}{\text{vol}(\mathcal{B}_n(1))^{2/n}} \leq 1 + \frac{n}{4},$$

where the last inequality comes from well known bounds on $\text{vol}(\mathcal{B}_n(1))$ [NV10, Chapter 2]. So given the volume of a lattice, we immediately have an upper bound on the length of its shortest vector. Unfortunately no similar bound for the other successive minimas is known. Minkowski's second theorem bounds the geometric mean of the first $d \leq n$ successive minimas (see e.g. [NV10, Chapter 2]):

$$\left(\prod_{i=1}^{d} \lambda_i \right)^{1/d} \leq \sqrt{\gamma_n} \det(\mathcal{L})^{1/n} \quad \text{for} \quad 1 \leq d \leq n.$$

While there exist lattices for which Minkowski's first theorem (1.1.3) is asymptotically tight ([MG02, Chapter 1.2]), lattices of fixed volume can be constructed in which the shortest vector is arbitrarily short.

There exist some heuristic on the lengths of the consecutive minimas of a lattice (see e.g. [NV10, Chapter 2], [GN08]).

- For a measurable subset $\mathbf{S} \subset \text{span}(\mathcal{L})$ we have $|\mathbf{S} \cap \mathcal{L}| \approx \frac{\text{vol}_n(\mathbf{S})}{\text{vol}(\mathcal{L})}$.
- Further $\lambda_i(\mathcal{L}) \approx \left(\frac{\text{vol}(\mathcal{L})}{\text{vol}_n\left(\mathcal{B}_n(1)\right)} \right)^{1/n}$.

They are commonly known as *Gaussian Heuristic*. The second heuristic can be seen as consequence from the first: Choose $\mathbf{S} = \mathcal{B}_m\big(\lambda_i(\mathcal{L})\big) \cap \text{span}(\mathcal{L})$. Then we expect $|\mathbf{S} \cap \mathcal{L}|$ to be independent of n and hence $|\mathbf{S} \cap \mathcal{L}| \approx \frac{\text{vol}_n(\mathbf{S})}{\text{vol}(\mathcal{L})} = \lambda_i^n \frac{\text{vol}_n(\mathcal{B}_n(1))}{\text{vol}(\mathcal{L})}$ implies the second heuristic.

1.2 Motivation

Lattices are objects of huge interest in both cryptography and cryptanalysis. Their role in cryptanalysis dates back to 1982, when Lenstra, Lenstra and Lovàsz [LLL82] presented the well-known LLL algorithm, that in polynomial time computes a reduced basis from an arbitrary lattice basis. Having a wide range of other applications from number theory to integer programming, it serves as strong cryptanalytic tool (e.g. [Odl90, JS94]). From the cryptography side of view, the interest in lattices is mostly due to some well studied and provable hard problems associated with them. We refer to e.g. [MG02] for an overview of complexity results on lattice problems. Probably the two most famous ones are the *shortest vector problem* (SVP) and the *closest vector problem* (CVP) and their approximate versions respectively. Let $\mathcal{L}$ be a lattice of rank n. For $\gamma \geq 1$, the γ-approximate SVP asks for a vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma\|\mathbf{w}\|$ for all $\mathbf{w} \in \mathcal{L}$. Similarly the γ-approximate CVP asks, given an arbitrary point $\mathbf{t} \in \text{span}(\mathcal{L})$, for a vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}-\mathbf{t}\| \leq \gamma\|\mathbf{w}-\mathbf{t}\|$, for all $\mathbf{w} \in \mathcal{L}$. If $\gamma = 1$, we simply talk about the SVP and CVP respectively. While the NP hardness of CVP for any norm $\|\cdot\|_p$, $p \in \mathbb{N} \cup \infty$, has been known since 1981, the hardness of SVP was only known to be NP hard for the infinity norm $\|\cdot\|_\infty$ by that time [vEB81]. In 1998, Ajtai managed to prove the NP hardness of SVP (with respect to the Euclidean norm) under randomized reductions [Ajt98] and it has been proven that SVP is even NP hard to approximate within a factor $\gamma < \sqrt{2}$ under randomized reductions [Mic01]. Also the approximate version of CVP has proven to be NP hard with an approximation factor $\gamma = 2^{\mathcal{O}(\log n/\log\log n)}$ [DKS98], which is exponentially smaller than n^c for any $c > 0$. However both problems are not likely to be NP hard to approximate with approximation factor $\gamma \geq \sqrt{n/\log n}$ [GG00]. Nevertheless, no polynomial time algorithm solving the approximate versions of SVP and CVP with polynomial approximation factor is known. The above mentioned LLL algorithm directly solves the approximate SVP with approximation factor $\gamma = 2^{\mathcal{O}(n)}$ in polynomial time. The LLL algorithm is also a key part of Babai's nearest plane algorithm [Bab86] which solves the approximate SVP with approximation factor $\gamma = 2^{\mathcal{O}(n)}$ in polynomial time.

A problem being NP hard only guarantees that it is hard in the worst-case, while it might be that most instances are easy to solve, i.e. it is easy on average. Basing cryptographic functions on these kinds of problems is clearly of limited use. Since the celebrated result by Ajtai [Ajt96] in 1996, who came up with a class of integer lattices for which the average-case complexity of the SVP is based on the worst-case complexity of some —approximate, with polynomial factor— lattice problems, lattices are a central object of study when it comes to the construction of provable secure cryptosystems. The security proofs of these constructions rely on the hardness of some approximate versions of lattice problems and therefore any improvement in lattice reduction algorithms has direct influence on the choice of the parameters of these constructions. Due to the gap between the provable NP hardness of lattice problems and what modern lattice basis reduction algorithms are capable of achieving, even remarkable improvements in lattice basis reduction would not make the complexity hierarchy collapse, i.e. imply that P=NP.

It is common knowledge that modern lattice reduction algorithms in practice perform

better than guaranteed by their provable worst case behaviour. Recently systematic empirical evidence [NS06, GN08] has been generated supporting this theory. It seems that modern lattice basis reduction algorithm achieve an approximation factor for SVP of the form α^n, where $\alpha > 1$. It is argued that while $\alpha \approx 1.012$ can be achieved, $\alpha \leq 1.005$ seems completely out of reach in dimensions beyond $n = 500$.

Supported by empirical evidence, the author of this thesis believe that heuristic methods could drastically improve the performance of modern lattice basis reduction algorithm. For example an algorithm basically repeatedly performing LLL reduction followed by ordering the basis vectors according to their lengths lead to entries in the hall of fame.[1]

1.3 Overview of results

In Chapter 2 we review the well known LLL reduction algorithm and its early improvements DeepLLL and BKZ. The main contribution is a new algorithm, called PotLLL, reducing the theoretical and practical gap between LLL and DeepLLL. We prove that the new algorithm runs in polynomial time and show that it outperforms LLL by means of the Hermite factor of the shortest basis vector in practice. To our knowledge our algorithm is the first improvement of LLL that still provably runs in deterministic polynomial time. We also compare its practical behaviour to BKZ with small blocksizes and a DeepLLL variant with small blocksizes. We further give an independent view on the practical performance of modern lattice basis reduction algorithms as has been conduced in [GN08, NS06]. Most results of this chapter are contained in [FSW13] and have been accepted for the International Workshop on Coding and Cryptography - WCC 2013.

In Chapter 3 we improve the technique to solve the CVP based on dual HKZ-bases by J. Blömer [Blö00]. His technique is based on the transference theorems given by Banaszczyk [Ban93] which imply some necessary conditions on the coefficients of the closest vectors with respect to a basis whose dual is HKZ-reduced. Recursively, starting with the last coefficient, intervals of length i can be derived for the i-th coefficient of any closest vector. This leads to $n! = e^{n \ln n + \mathcal{O}(n)}$ candidates for closest vectors. In this chapter we refine the necessary conditions derived from the transference theorems, giving an exponential reduction of the number of candidates. Our improvement is due to the fact that the lengths of the intervals are not independent. In the original algorithm the candidates for a coefficient pair (v_i, v_{i+1}) correspond to the integer points in a rectangle of volume $i \cdot (i+1)$. In our analysis we show that the candidates for (v_i, v_{i+1}) in fact lie in an ellipse with transverse and conjugate diameter $i+1$, respectively i. This reduces the expected number of points to be enumerated by an exponential factor of about 0.886^n. We further show how a choice of the coefficients $(v_{i+1}, \ldots, v_n)$ influences the interval from which v_i can be chosen. Numerical computations show that these considerations allow to bound the expected number of points to be enumerated by $n^{0.75n} = e^{0.75n \ln n}$

[1] `http://www.latticechallenge.org/svp-challenge/index.php`

for $10 \leq n \leq 2000$. The results of this chapter are contained in [WM12] and have been submitted for publication.

In Chapter 4, a probabilistic algorithm by Ajtai et al. [AKS01] solving the shortest vector problem with $2^{\mathcal{O}(n)}$ time and space complexity is at the center of our considerations. We derive new bounds on the number of points with minimal mutual distance inside a ball of given radius and derive new complexity upper bounds on the aforementioned algorithm. We improve the complexity upper bound by Nguyen et al. [NV08] from time $2^{5.9n+\mathcal{O}(\log n)}$ and space $2^{2.95n+\mathcal{O}(\log n)}$ to time $2^{4.2n+\mathcal{O}(\log n)}$ and space $2^{2.1n+\mathcal{O}(\log n)}$. Further we generalize the heuristic sieving algorithm proposed in [NV08] in a way that we think could close the gap between sieving techniques and enumeration. Instead of considering only pairs of lattice vectors in the sieving steps, we propose to run a SVP-solver on β-tuples, $2 \leq \beta \leq n$, of lattice vectors. First steps of the analysis thereof are taken, a complete analysis is still to be done.

The starting point to the considerations in Chapter 5 was a new way to express the Seysen measure of a lattice basis as trace of a symmetric positive definite matrix by Maze [Maz10]. The Seysen measure equals the sum of the corresponding eigenvalues having harmonic mean 1. The orthogonality defect, another way of measuring the reducedness of a basis, corresponds to the product of these eigenvalues. This motivated the work on bounds on the harmonic, geometric mean respectively, when the two other means are known. Our main contributions are a tight upper and lower bound on the geometric mean when the arithmetic and harmonic means are known (Theorem 5.2.8) and similarly an upper and lower bound on the harmonic mean when both the geometric and arithmetic means are known (Theorem 5.2.7). These new inequalities imply new inequalities between the Seysen measure and the orthogonality defect of a lattice basis.

In the first part of the chapter we further bound the distance of coefficient vectors by the distance of the according vectors and the Seysen measure of the respective basis (Lemma 5.1.4). The results on the harmonic, geometric and arithmetic means inequalities have been published in [MW12].

The initial motivation for Chapter 6 was the development of a genetic algorithm to improve existing lattice reduction algorithms. Having a population of fairly reduced lattice bases, we were looking for ways to combine these bases to generate a new population of further reduced basis. For lattices of rank n the idea was to take a subset of $k_1 < n$ basis vectors from one basis and a subset of $k_2 < n - k_1$ basis vectors from a second basis, and complete them into a new basis containing the chosen basis vectors. The question was whether we can expect this kind of recombination to be successful with reasonable probability. The result of this chapter gives a partial answer to the question. Under the assumption that the coefficients of the chosen basis vectors with respect to some fixed basis are uniformly at random from a box $[-\beta, \beta]^n \cap \mathbb{Z}^n$, we compute the probability that it is possible to recombine two bases in this way for $\beta \to \infty$. More generally, we show (Proposition 6.2.5) that the natural density of unimodular $k \times n$ matrices for $k < n$,

equals $\mathcal{D}_{k,n} := \prod_{i=n-k+1}^{n} \zeta(i)^{-1}$, where $\zeta(\cdot)$ is the well known Riemann zeta function. The considerations made also give an answer to the dual version of the above described problem: Given $l > n$ vectors such that the corresponding coefficients are uniformly at random from a large box, the probability that these vectors generate the lattice tends to $\mathcal{D}_{n,l} := \prod_{i=l-n+1}^{l} \zeta(i)^{-1}$.

Unfortunately the genetic algorithm as mentioned above did not prove to give a considerable improvement of existing lattice basis reduction algorithms. However the results on the natural density of unimodular matrices has been published in [MRW11].

Chapter 2

Improvements of LLL

In 1982, Lenstra, Lenstra and Lovàsz [LLL82] came up with a notion of a reduced lattice basis together with a polynomial time algorithm to compute such bases. It is normally referred to as the LLL reduction algorithm and the according basis is called LLL-reduced. It is considered to be one of the most important algorithmic achievements in the last century [NV10, Preface] and it has applications in a wide area reaching from number theory [NV10, Chapter 7] to integer programming [NV10, Chapter 9]. It has further already at an early stage proved to be a powerful tool for cryptanalysis. As such it can for example be used to solve some instances of the subset sum problem [LO85] directly leading to attacks to knapsack type of cryptosystems (e.g. [MH78]). As lattices remain popular when it comes to constructing provably secure cryptosystems, any improvement of the existing algorithms is of interest to the crypto-community. Probably the most important improvements of LLL are due to Schnorr and Euchner, who suggest the so called BKZ algorithm and LLL with deep insertions (DeepLLL) [SE94]. While the deep insertion variant of the LLL algorithm in practice finds shorter vectors that the original LLL algorithm, its running time is no longer polynomial. In this chapter we present the PotLLL algorithm [FSW13], which outputs a basis that satisfies stronger conditions than LLL, but weaker conditions than DeepLLL. We will show that neither DeepLLL nor PotLLL satisfy stronger bounds than those proven for LLL when it comes to the length of the first basis vector. However there is strong empirical evidence that the proven worst case bounds for the first basis vector of reduced lattice bases do not reflect the actual performance of the reduction algorithms [NS06, GN08]. We compare the practical behaviour of our new algorithm to different variants of LLL, DeepLLL and BKZ. We will see that our new algorithm in practice leads to shorter vectors than the original LLL algorithm. As our algorithm still runs in provable polynomial time, we consider it a serious alternative to the classical LLL algorithm. Further we reproduce and extend the results in [NS06, GN08] on the practical behaviour of the different reduction algorithms.

We start by defining the potential of a lattice basis and its behaviour under certain (unimodular) operations on the lattice basis in Section 2.1. This will play a main role throughout the chapter. We then quickly review the definition of an LLL reduced basis and the principle of deep insertions which was an early improvement of the original

algorithm and give the definition of a BKZ reduced basis. In Section 2.4 we introduce the new notion of a PotLLL reduced basis and give an algorithm to compute it out of an arbitrary lattice basis in Section 2.4.1. We further give empirical results on the performance of PotLLL and compare it to different variants of LLL, DeepLLL and BKZ. Finally in Section 2.5 possible extension for future work are suggested.

Most of the results in this chapter emerged out of joint work with Felix Fontein and Michael Schneider and is submitted [FSW13].

2.1 The potential of a lattice basis

In this section we define the potential of a lattice basis as it is used in the proof of the polynomial running time of LLL [LLL82]. If we see the vectors of a basis B as columns of a matrix, we show how the potential of the basis changes under certain elementary column operations. These observations make it easy to understand the polynomial running time of LLL and will be important in the discussion of our new polynomial time version of LLL with deep insertions.

Definition 2.1.1 *The* potential $\mathrm{Pot}(B)$ *of a lattice basis* $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ *is defined as*

$$\mathrm{Pot}(B) := \prod_{i=1}^{n} \det\left(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)\right)^2 = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|^{2(n-i+1)}.$$

The potential, unlike the determinant depends on the ordering of the lattice basis. We will now define a special family of permutations on the basis vectors, and then examine how the potential changes under these permutations. Let S_n denote the group of permutations of n elements. By applying $\sigma \in S_n$ to a basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, the basis vectors are reordered, i.e. $\sigma B = [\mathbf{b}_{\sigma(1)}, \ldots, \mathbf{b}_{\sigma(n)}]$.

For $1 \leq k \leq l \leq n$ let us define a family of elements $\sigma_{k,l} \in S_n$ as follows:

$$\sigma_{k,l}(i) = \begin{cases} i & \text{for} \quad i < k \text{ or } i > l, \\ l & \text{for} \quad i = k, \\ i-1 & \text{for} \quad k < i \leq l. \end{cases} \tag{2.1.1}$$

The following example illustrates the permutation:

Example 2.1.2 *Let* $1 \leq k \leq l \leq n$ *and* $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$*, then*

$$\begin{array}{rcl} B & = & [\ \mathbf{b}_1 \ \ldots \ \mathbf{b}_{k-1} \ \mathbf{b}_k \ \mathbf{b}_{k+1} \ \ldots \ \ldots \ \mathbf{b}_{l-1} \ \mathbf{b}_l \ \mathbf{b}_{l+1} \ \ldots \ \mathbf{b}_n \ ], \\ \sigma_{k,l}B & = & [\ \mathbf{b}_1 \ \ldots \ \mathbf{b}_{k-1} \ \mathbf{b}_l \ \mathbf{b}_k \ \mathbf{b}_{k+1} \ \ldots \ \ldots \ \mathbf{b}_{l-1} \ \mathbf{b}_{l+1} \ \ldots \ \mathbf{b}_n \ ]. \end{array}$$

Note that $\sigma_{k,l} = \sigma_{k,k+1}\sigma_{k+1,k+2}\cdots\sigma_{l-1,l}$ and that $\sigma_{k,k+1}$ is swapping the two elements at position k, $k+1$ respectively. The potential is hereby changed as follows:

Lemma 2.1.3 *Given a lattice basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. Then for $1 \leq k < n$,*

$$\mathrm{Pot}(\sigma_{k,k+1}B) = \frac{\|\pi_k(\mathbf{b}_{k+1})\|^2}{\|\pi_k(\mathbf{b}_k)\|^2}\mathrm{Pot}(B).$$

Proof: Note that for $j \neq k$,

$$\det\big(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_j)\big) = \det\big(\mathcal{L}(\mathbf{b}_{\sigma_{k,k+1}(1)}, \ldots, \mathbf{b}_{\sigma_{k,k+1}(j)})\big).$$

Hence, as $\det\big(\mathcal{L}(\mathbf{b}_{\sigma_{k,k+1}(1)}, \ldots, \mathbf{b}_{\sigma_{k,k+1}(k)})\big) = \|\pi_k(\mathbf{b}_{k+1})\| \prod_{1\leq i\leq k-1} \|\mathbf{b}_i^*\|$ we get

$$\frac{\mathrm{Pot}(\sigma_{k,k+1}B)}{\mathrm{Pot}(B)} = \frac{\det\big(\mathcal{L}(\mathbf{b}_{\sigma_{k,k+1}(1)}, \ldots, \mathbf{b}_{\sigma_{k,k+1}(k)})\big)^2}{\det\big(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_k)\big)^2} = \frac{\|\pi_k(\mathbf{b}_{k+1})\|^2}{\|\pi_k(\mathbf{b}_k)\|^2}.$$

□

It is now not hard to see how the potential of a lattice basis changes under arbitrary permutations in the family defined in (2.1.1):

Corollary 2.1.4 *Let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be a lattice basis. Then for $1 \leq k \leq l \leq n$*

$$\mathrm{Pot}(\sigma_{k,l}B) = \mathrm{Pot}(B) \prod_{i=k}^{l} \frac{\|\pi_i(\mathbf{b}_l)\|^2}{\|\pi_i(\mathbf{b}_i)\|^2}.$$

Proof: The proof is by induction over k. The claim is true for $k = l$. For $k < l$, $\sigma_{k,l} = \sigma_{k,k+1}\sigma_{k+1,l}$. As $\mathbf{b}_l$ is the $(k+1)$th basis vector of $\sigma_{k+1,l}B$, with Lemma 2.1.3 we get

$$\mathrm{Pot}(\sigma_{k,l}B) = \mathrm{Pot}(\sigma_{k,k+1}\sigma_{k+1,l}B) = \frac{\|\pi_k(\mathbf{b}_l)\|^2}{\|\pi_k(\mathbf{b}_k)\|^2}\mathrm{Pot}(\sigma_{k+1,l}B),$$

which finishes the proof. □

Permutations as defined in (2.1.1) will be one of the two kinds of operations applied to a lattice basis during both the LLL algorithm and PotLLL. The second is, for $1 \leq k < l \leq n$, adding an integer multiple of $\mathbf{b}_k$ to $\mathbf{b}_l$. However this does not change the potential of the respective basis:

Lemma 2.1.5 *Let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be the basis of a lattice. For $1 \leq k < l \leq n$ and $x \in \mathbb{Z}$, let*

$$B' = [\mathbf{b}'_1, \ldots, \mathbf{b}'_n],$$

such that $\mathbf{b}'_i = \mathbf{b}_i$ for $i \neq l$ and $\mathbf{b}'_l = \mathbf{b}_l + x\mathbf{b}_k$. Then $B'^* = B^*$ and in particular,

$$\mathrm{Pot}(B') = \mathrm{Pot}(B).$$

Proof: Trivially $\pi_i(\mathbf{b}_i) = \pi_i(\mathbf{b}'_i)$ for $i < l$. Further by Equation (1.1.1) and (1.1.2)

$$\begin{aligned}\pi_l(\mathbf{b}'_l) &= \pi_l(\mathbf{b}_l + x\mathbf{b}_k) = \mathbf{b}_l + x\mathbf{b}_k - \sum_{j=1}^{l-1} \frac{\langle \mathbf{b}_l + x\mathbf{b}_k, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \mathbf{b}_j^* \\ &= \mathbf{b}_l - \sum_{j=1}^{l-1} \frac{\langle \mathbf{b}_l, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \mathbf{b}_j^* + x\mathbf{b}_k - x \sum_{j=1}^{l-1} \frac{\langle \mathbf{b}_k, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \mathbf{b}_j^* = \pi_l(\mathbf{b}_l),\end{aligned}$$

and hence also for $i > l$, $\pi_i(\mathbf{b}_i) = \pi_i(\mathbf{b}'_i)$. □

2.2 LLL

The LLL lattice basis reduction algorithm [LLL82] is probably the most famous algorithm for lattice reduction, computing a LLL reduced basis. See Algorithm 1 on page 17 for a high level description the algorithm.

Definition 2.2.1 *Let $\delta \in (1/4, 1]$. A basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called* δ-LLL reduced *if and only if it satisfies the following two conditions*

1. $\forall 1 \leq j < i \leq n : |\mu_{i,j}| \leq \frac{1}{2}$ *(size-reduced).*
2. $\forall 1 \leq k < n : \delta \cdot \|\pi_k(\mathbf{b}_k)\|^2 \leq \|\pi_k(\mathbf{b}_{k+1})\|^2$ *(Lovász-condition).*

Remark 2.2.2 *Using the notation as above, by Lemma 2.1.3, the Lovász-condition can equivalently be stated as*

$$\forall 1 \leq k < n : \delta \cdot \mathrm{Pot}(B) \leq \mathrm{Pot}(\sigma_{k,k+1} B)$$

For $\delta < 1$, a δ-LLL reduced basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ can be computed in polynomial time in the input size. It is proven to approximate the shortest vector of a lattice up to an exponential factor:

$$\|\mathbf{b}_1\| \leq \left(\frac{1}{\delta - 1/4}\right)^{\frac{n-1}{2}} \lambda_1(\mathcal{L}(B)), \tag{2.2.4}$$

$$\|\mathbf{b}_1\| \leq \left(\frac{1}{\delta - 1/4}\right)^{\frac{n-1}{4}} \mathrm{vol}(\mathcal{L}(B))^{1/n}. \tag{2.2.5}$$

It is clear that these bounds are stronger the closer δ is to 1. For $\delta = 1$, these bounds can be shown to be tight. In fact there exist so called *critical bases*, i.e. bases which are LLL reduced and reach these bounds with equality [Sch94]. However in practice, the LLL algorithm behave much better than guaranteed by these worst case bounds as has been shown in [NS06, GN08]. In fact, the experiments in [NS06, GN08] indicate that δ-LLL reduction with reduction parameter $\delta = 0.99$ achieve

$$\|\mathbf{b}_1\| \leq c^n \mathrm{vol}(\mathcal{L}(B))^{1/n},$$

with $c \approx 1.02$.

2.2.1 Deep insertions

One early attempt to improve the LLL reduction algorithms is due to Schnorr and Euchner [SE94] who came up with the notion of a DeepLLL reduced basis.

Definition 2.2.3 *Let $\delta \in (1/4, 1]$. A basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called* δ-DeepLLL reduced with blocksize β *if and only if it satisfies the following two conditions*

1. $\forall 1 \leq j < i \leq n : |\mu_{i,j}| \leq \frac{1}{2}$ *(size-reduced).*
2. $\forall 1 \leq k < l \leq n$ *with* $k \leq \beta \vee l - k \leq \beta : \delta \cdot \|\pi_k(\mathbf{b}_k)\|^2 \leq \|\pi_k(\mathbf{b}_l)\|^2$.

For $\beta = n$ we simply call this a DeepLLL reduced basis. For a high level description of a DeepLLL reduction algorithm see Algorithm 2 on page 17. It is clear that a DeepLLL reduced basis is also LLL reduced. As such the first basis vector of a δ-DeepLLL reduced basis also satisfies the bounds given in (2.2.4) and (2.2.5). The question is whether DeepLLL reduced bases satisfy stronger bounds. The answer is negative in the case of (2.2.5) and reduction parameter $\delta = 1$. Consider the following adaption of the critical basis with respect to LLL [Sch94]:

$$A_n(\alpha) := \begin{pmatrix} 1 & \frac{1}{2} & \cdots & & \cdots & \frac{1}{2} \\ 0 & \alpha & \frac{\alpha}{2} & \cdots & \cdots & \frac{\alpha}{2} \\ \vdots & \ddots & \alpha^2 & \frac{\alpha^2}{2} & \cdots & \frac{\alpha^2}{2} \\ & & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \alpha^{n-2} & \frac{\alpha^{n-2}}{2} \\ 0 & \cdots & & \cdots & 0 & \alpha^{n-1} \end{pmatrix}. \tag{2.2.6}$$

The following proposition shows that this basis is critical with respect to 1-DeepLLL.

Proposition 2.2.4 *For $\alpha = \sqrt{3/4}$, the column vectors of $A_n(\alpha)$ define a δ-DeepLLL reduced basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ with $\delta = 1$ and $\|\mathbf{b}_1\|^2 = \frac{1}{\alpha^{(n-1)/2}} \mathrm{vol}\big(\mathcal{L}(A_n)\big)^{1/n}$.*

Proof: From the diagonal form of A_n it is easy to see that

$$\mathrm{vol}(\mathcal{L}) = \det(A_n) = \alpha^{n(n-1)/2}.$$

Hence $\|\mathbf{b}_1\| = 1 = \alpha^{-(n-1)/2}\mathrm{vol}(\mathcal{L})^{1/n}$. It remains to show that A_n is DeepLLL reduced. For all $1 \leq j < i \leq n$ we have

$$\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} = \frac{\alpha^{(j-1)/2}/2}{\alpha^{(j-1)/2}} = \frac{1}{2}.$$

and

$$\|\pi_j(\mathbf{b}_i)\|^2 = \alpha^{2(i-1)} + \frac{1}{4}\sum_{l=j}^{i-1} \alpha^{2(l-1)} = \alpha^{2(j-1)} \left(\frac{1}{4} \sum_{l=0}^{i-j-1} \alpha^{2l} + \alpha^{2(i-j)} \right).$$

For $\alpha = \sqrt{3/4}$,

$$\frac{1}{4}\sum_{l=0}^{i-j-1}\alpha^{2l} + \alpha^{2(i-j)} = \frac{1}{4}\frac{1-\alpha^{2(i-j)}}{1-\alpha^2} + \alpha^{2(i-j)} = 1,$$

giving $\|\pi_j(\mathbf{b}_i)\|^2 = \alpha^{2(j-1)} = \|\pi_j(\mathbf{b}_j)\|^2$. □

While we can not prove any stronger bounds on the length of the first basis vector for DeepLLL reduced bases than for LLL reduced bases, algorithms to compute DeepLLL reduced bases give shorter vectors in practice [GN08]. However no polynomial time algorithm is known to compute DeepLLL reduced bases. The authors of [SE94] claim that a DeepLLL reduced basis with low blocksize can be computed in polynomial time, however we are not aware of any proof thereof. This led to PotLLL, a weaker version of DeepLLL reduction that provably can be computed in polynomial time (see Section 2.4).

2.3 BKZ

In this section we will quickly introduce the notion of a block Korkine-Zolotarev (BKZ) reduced basis [Sch87] which can be seen as generalization of LLL to higher blocksizes. Consider the lattice $\mathcal{L}\big(\pi_k(\mathbf{b}_k), \pi_k(\mathbf{b}_{k+1})\big)$ generated by $\pi_k(\mathbf{b}_k), \pi_k(\mathbf{b}_{k+1})$ for $1 \leq k < n$. Then for $\delta = 1$ the two conditions in Definition 2.2.3 are equivalent with the fact that $\pi_k(\mathbf{b}_k), \pi_k(\mathbf{b}_{k+1})$ is Gauss reduced and in particular $\|\pi_k(\mathbf{b}_k)\| = \lambda_1\big(\mathcal{L}(\pi_k(\mathbf{b}_k), \pi_k(\mathbf{b}_{k+1})\big)$ and $\|\pi_k(\mathbf{b}_{k+1})\| = \lambda_2\big(\mathcal{L}(\pi_k(\mathbf{b}_k), \pi_k(\mathbf{b}_{k+1})\big)$ [MG02, Chapter 2]. The idea of the BKZ reduction is to extend this requirements to larger blocks, i.e.

$$\|\pi_k(\mathbf{b}_k)\| = \lambda_1\big(\mathcal{L}(\pi_k(\mathbf{b}_k), \ldots, \pi_k(\mathbf{b}_l)\big),$$

with $l = k + \beta - 1$ for some $\beta \geq 2$. In order to achieve this, the BKZ basis reduction algorithm includes an enumeration routine.

We will not further discuss the algorithm computing BKZ reduced bases. We restrict ourselves to giving the formal definition and some properties of a BKZ reduced basis. The interested reader is referred to [SE94] or for a state of the art description of BKZ reduction to [CN11].

Definition 2.3.1 *Let $\delta \in (1/4, 1]$. A basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called* δ-BKZ reduced *with blocksize β if and only if it satisfies the following two conditions*

1. $\forall 1 \leq j < i \leq n : |\mu_{i,j}| \leq \frac{1}{2}$ *(size-reduced).*
2. $\forall 1 \leq k < n : \delta \cdot \|\pi_k(\mathbf{b}_k)\|^2 \leq \|\lambda_1(\pi_k(\mathcal{L}(\mathbf{b}_k, \ldots, \mathbf{b}_{\min\{n, k+\beta-1\}}))\|^2$.

Remark 2.3.2 *For $\delta = 1$, the definition of a BKZ reduced basis with blocksize $\beta = 2$ coincides with the definition of a 1-LLL reduced basis and for $\beta = n$ it coincided with the definition of a HKZ basis.*

A BKZ reduced basis with blocksize β provably satisfies the following bounds [Sch94, GN08]

$$\|\mathbf{b}_1\| \leq \gamma_\beta^{(n-1)/(\beta-1)} \lambda_1(\mathcal{L}),$$
$$\|\mathbf{b}_1\| \leq \sqrt{\gamma_\beta}^{(n-1)/(\beta-1)+1} \mathrm{vol}(\mathcal{L})^{1/n}.$$

where γ_β is the Hermite constant as in Section 1.1. It is clear that algorithms to compute BKZ with blocksize β have complexity exponential in β. While BKZ reduction for $\beta \leq 20$ can be done in reasonable time, no useful complexity upper bounds are known for these algorithms. Clearly the deep insertion principle can also be applied to BKZ reduction.

2.4 PotLLL

In this section we present a variant of DeepLLL that runs in polynomial time. This is joint work with Felix Fontein and Michael Schneider [FSW13]. We start by defining the notion of a PotLLL reduced basis. We further give an algorithm for computing such a basis together with a proof of the polynomial running time thereof and empirically compare our algorithm to LLL, DeepLLL and BKZ for different blocksizes.

Definition 2.4.1 *Let $\delta \in (1/4, 1]$. A lattice basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is* δ-PotLLL reduced *if and only if*

1. $\forall 1 \leq j < i \leq n : |\mu_{i,j}| \leq \frac{1}{2}$ *(size-reduced).*
2. $\forall 1 \leq k < l \leq n : \delta \cdot \mathrm{Pot}(B) \leq \mathrm{Pot}(\sigma_{k,l}(B))$.

The following two lemmas show that PotLLL reducedness is intermediate between LLL reducedness and DeepLLL reducedness.

Lemma 2.4.2 *A δ-PotLLL reduced basis B is also δ-LLL reduced.*

Proof: It is enough to show that the Lovász-condition is equivalent to the second condition in Definition 2.4.1 restricted to consecutive pairs, i.e. $l = k + 1$. So we have to show that the following equivalence:

$$\delta \cdot \mathrm{Pot}(B) \leq \mathrm{Pot}(\sigma_{i,i+1}(B)) \Leftrightarrow \delta \|\pi_i(\mathbf{b}_i)\|^2 \leq \|\pi_i(\mathbf{b}_{i+1})\|^2.$$

This immediately follows from Lemma 2.1.3. □

Hence the PotLLL reduction is at least as strong as the LLL reduction.

Lemma 2.4.3 *For $\delta \in \left(4^{-1/(n-1)}, 1\right]$, a δ-DeepLLL reduced basis B is also δ^{n-1}-PotLLL reduced.*

Proof: We proceed by contradiction. Assume that B is not δ^{n-1}-PotLLL reduced, i.e. there exist k, l with $1 \leq k < l \leq n$ such that

$$\delta^{n-1}\mathrm{Pot}(B) > \mathrm{Pot}(\sigma_{k,l}B).$$

By Corollary 2.1.4 this is equivalent to

$$\delta^{n-1} > \prod_{i=k}^{l} \frac{\|\pi_i(\mathbf{b}_l)\|^2}{\|\pi_i(\mathbf{b}_i)\|^2} = \prod_{i=k}^{l-1} \frac{\|\pi_i(\mathbf{b}_l)\|^2}{\|\pi_i(\mathbf{b}_i)\|^2}.$$

It follows that there exist k, j, l with $k \leq j < l$ such that $\|\pi_j(\mathbf{b}_l)\|^2/\|\pi_j(\mathbf{b}_j)\|^2 < \delta^{(n-1)/(l-k)} < \delta$ which implies that B is not δ-DeepLLL reduced. □

So for $\delta = 1$, a DeepLLL reduced lattice basis is also PotLLL reduced. This relation immediately implies that the first basis vector of a δ-PotLLL reduced basis satisfies (2.2.5). On the other hand we have seen in Section 2.2.1 that for $\delta = 1$ no stronger bounds concerning the length of the first basis vector of a DeepLLL reduced basis exist, than those inherited from LLL bases. As a consequence, no stronger bounds exist for PotLLL reduced bases either.

Corollary 2.4.4 *For $\alpha = \sqrt{3/4}$, the row vectors of $A_n(\alpha)$ as defined in* (2.2.6) *define a δ-PotLLL reduced basis with $\delta = 1$ and $\|\mathbf{b}_1\|^2 = \frac{1}{\alpha^{(n-1)/2}}\mathrm{vol}(\mathcal{L}(A_n))^{1/n}$.*

In the next section we show how a PotLLL reduced basis can be computed in polynomial time from an arbitrary lattice basis.

2.4.1 The PotLLL reduction algorithm

The original LLL algorithm basically performs two kinds on operations on the lattice basis. One is the size-reduction, ensuring that the output basis is size-reduced. The basis is hereby changed as in Lemma 2.1.5. Consequently the potential is not affected by this operation. The other one is the swapping of two adjacent basis vectors if they do not satisfy the Lovász-condition. These swappings reduce the potential of the basis by a factor at least δ. As the potential is lower bounded $\mathrm{vol}(\mathcal{L})$ the number of swaps can be bounded giving an upper bound on the number of loop iterations. This is basically how the proof of the polynomial running time of LLL works. In the DeepLLL reduction algorithm however, it might happen that a basis vector is 'deep inserted' at a position such that the potential of the basis increases. This is why the algorithm no longer runs in polynomial time. The main idea of PotLLL now is to allow deep insertions as in DeepLLL, however only under the condition that the deep insertion results in a decrease of the potential of the basis by a factor of at least δ. See Algorithm 3 (page 17) for a high level description of the PotLLL reduction. The similarity to the classical LLL algorithm (Algorithm 1) and the classical DeepLLL reduction (Algorithm 2) can be seen immediately.

Algorithm 1: LLL

Input: Basis $B \in \mathbb{Z}^{m \times n}$, $\delta \in (1/4, 1]$
Output: A δ-LLL reduced basis.

$l \leftarrow 2$
while $l \leq n$ **do**
  Size-reduce(B)
  $k \leftarrow l - 1$
  if $\delta \cdot \|\pi_k(\mathbf{b}_k)\|^2 > \|\pi_k(\mathbf{b}_l)\|^2$ **then**
    $B \leftarrow \sigma_{k,l} B$
    $l \leftarrow k$
  else
    $l \leftarrow l + 1$
  end
end
return B

Algorithm 2: DeepLLL

Input: Basis $B \in \mathbb{Z}^{m \times n}$, $\delta \in (1/4, 1]$
Output: A δ-DeepLLL reduced basis.

$l \leftarrow 2$
while $l \leq n$ **do**
  Size-reduce(B)
  $k \leftarrow \operatorname{argmin}_{1 \leq j \leq l} \|\pi_j(\mathbf{b}_l)\|$
  if $\delta \cdot \|\pi_k(\mathbf{b}_k)\|^2 > \|\pi_k(\mathbf{b}_l)\|^2$ **then**
    $B \leftarrow \sigma_{k,l} B$
    $l \leftarrow k$
  else
    $l \leftarrow l + 1$
  end
end
return B

Algorithm 3: PotLLL

Input: Basis $B \in \mathbb{Z}^{m \times n}$, $\delta \in (1/4, 1]$
Output: A δ-PotLLL reduced basis.

$l \leftarrow 2$
while $l \leq n$ **do**
  Size-reduce(B)
  $k \leftarrow \operatorname{argmin}_{1 \leq j \leq l} \mathrm{Pot}(\sigma_{j,l} B)$
  if $\delta \cdot \mathrm{Pot}(B) > \mathrm{Pot}(\sigma_{k,l} B)$ **then**
    $B \leftarrow \sigma_{k,l} B$
    $l \leftarrow k$
  else
    $l \leftarrow l + 1$
  end
end
return B

The proposition shows that for $\delta < 1$ the number of operations in the PotLLL algorithm is polynomially bounded in the dimension and the logarithm of the input size.

Proposition 2.4.5 *Let $\delta \in (1/4, 1)$ and $C = \max_{i=1,\dots,n} \|\mathbf{b}_i\|^2$. Then Algorithm 3 performs $\mathcal{O}\left(n^3 \log_{1/\delta}(C)\right)$ iterations of the* `while-loop` *in line 2.*

Proof: Note that in each iteration either the running index l is increased by one leaving the potential unchanged, or the potential of basis is reduced by a factor at least δ. It is clear from the definition of the potential of a basis, that it is lower bounded by the invariant $\mathrm{vol}(\mathcal{L})$. So let us upper bound the number of iterations N where a permutation happens. We start by upper bounding the potential I of the input basis. Note that with $\|\mathbf{b}_i^*\|^2 \leq \|\mathbf{b}_i\|^2 \leq C$ it follows that

$$d_j := \mathrm{vol}(\mathcal{L}(\mathbf{b}_1, \dots, \mathbf{b}_j))^2 = \prod_{i=1}^{j} \|\mathbf{b}_i^*\|^2 \leq C^j.$$

Consequently the potential I of the input basis satisfies

$$\prod_{j=1}^{n-1} d_j \cdot \mathrm{vol}(\mathcal{L}) \leq \prod_{j=1}^{n-1} C^j \cdot \mathrm{vol}(\mathcal{L}) \leq C^{\frac{n(n-1)}{2}} \cdot \mathrm{vol}(\mathcal{L}).$$

As the potential decreases by a factor δ after each permutation, the potential I_N of the basis after N permutations satisfies $I_N \leq \delta^N I$. However the potential is lower bounded by $\mathrm{vol}(\mathcal{L})$ and consequently N satisfies $\mathrm{vol}(\mathcal{L}) \leq \delta^N I$ and equivalently

$$N \leq \log_{1/\delta}(I/\mathrm{vol}(\mathcal{L})) \leq n(n-1)/2 \log_{1/\delta} C.$$

Now the number M of iterations where l is increased by 1 can be upper bounded by $(N+1)(n-1)$. Hence,

$$N + M = \mathcal{O}\left(n^3 \log_{1/\delta} C\right).$$

□

Technical details

We have seen that the number of loop operations in the PotLLL reduction algorithm is upper bounded by $\mathcal{O}\left(n^3 \log_{1/\delta} \max \|\mathbf{b}_i^*\|^2\right)$. In order to make a more precise statement on the running time we have to examine the operations done inside the `while-loop`. For this consider a more detailed code of the PotLLL algorithm given in Algorithm 4. Note that at the beginning of every iteration of the `while` loop, $\mathbf{b}_1, \dots, \mathbf{b}_{l-1}$ is PotLLL reduced. So specially they are size-reduced. Further the values $\|\mathbf{b}_j^*\|^2$ and $\mu_{i,j}$ are stored in memory for $1 \leq j < i \leq l-1$. Size reduction in line 3 performs the following operation

$$\mathbf{b}_l \leftarrow \mathbf{b}_l - \sum_{j=1}^{l-1} \lfloor \mu_{l,j} \rceil \, \mathbf{b}_j \quad \text{where} \quad \mu_{l,j} = \frac{\langle \mathbf{b}_l, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}.$$

Algorithm 4: Potential LLL, detailed version

Input: Basis $B \in \mathbb{Z}^{n\times n}$, $\delta \in (1/4, 1]$
Output: A δ-PotLLL reduced basis.

1 $l \leftarrow 2$
2 **while** $l \leq n$ **do**
3 Size-reduce($\mathbf{b}_l$ by $\mathbf{b}_1, \ldots, \mathbf{b}_{l-1}$)
4 Update($\|b_l^*\|^2$ and $\mu_{l,j}$ for $1 \leq j < l$)
5 $P \leftarrow 1, \quad P_{\min} \leftarrow 1, \quad k \leftarrow 1$
6 **for** $j = l-1$ *down to* 1 **do**
7 $P \leftarrow P \cdot \frac{\|\mathbf{b}_l^*\|^2 + \sum_{i=j}^{l-1} \mu_{l,i}^2 \|\mathbf{b}_i^*\|^2}{\|\mathbf{b}_j^*\|^2}$
8 **if** $P < P_{\min}$ **then**
9 $k \leftarrow j$
10 $P_{\min} \leftarrow P$
11 **end**
12 **end**
13 **if** $\delta > P_{\min}$ **then**
14 $B \leftarrow \sigma_{k,l} B$
15 Update($\|\mathbf{b}_k^*\|^2$ and $\mu_{k,j}$ for $1 \leq j < k$)
16 $l \leftarrow k$
17 **else**
18 $l \leftarrow l + 1$
19 **end**
20 **end**
21 **return** B

Using the Gram-Schmidt orthogonalization as described in [NS05, Figure 5] This results in a cost of $\mathcal{O}(mn)$ arithmetic operations in line 3 and 4.

In order to compute $\operatorname{argmin}_{1\leq j\leq l} \mathrm{Pot}(\sigma_{j,l}B)$ it is not necessary to compute the respective potentials. With

$$P_{j,l} := \frac{\mathrm{Pot}(\sigma_{j,l}B)}{\mathrm{Pot}(B)},$$

one can compute $\operatorname{argmin}_{1\leq j\leq l} P_{j,l}$ instead. Using $P_{l,l} = 1$ and

$$P_{j,l} = \frac{\mathrm{Pot}(\sigma_{j,l}B)}{\mathrm{Pot}(B)} = P_{j+1,l} \cdot \frac{\|\pi_j(\mathbf{b}_l)\|^2}{\|\pi_j(\mathbf{b}_j)\|^2} = P_{j+1,l} \cdot \frac{\|\mathbf{b}_l^*\|^2 + \sum_{i=j}^{l-1} \mu_{l,i}^2 \|\mathbf{b}_i^*\|^2}{\|\mathbf{b}_j^*\|^2}$$

for $j < l$ (Lemma 2.1.3), we can quickly determine $k = \operatorname{argmin}_{1\leq j\leq l} P_{j,l}$. The condition $\delta \cdot \mathrm{Pot}(B) > \mathrm{Pot}(\sigma_{k,l}B)$ then becomes $\delta > P_{k,l}$. Recalling that the values of $\|\mathbf{b}_j^*\|^2$ and $\mu_{l,j}$, $1 \leq j \leq l$, are stored in memory computing $P_{k,l}$ from $P_{l,l}$ can be done in $\mathcal{O}(n^2)$ arithmetic operations (`for-loop` starting on line 6). If a permutation happens, again $\|\mathbf{b}_k^*\|^2$ and $\mu_{k,j}$ have to be recomputed for $1 \leq j < k$ costing $\mathcal{O}(mn)$ arithmetic

operations. So the overall cost of one iteration is $\mathcal{O}(mn)$. We can conclude

Proposition 2.4.6 *Let $\delta \in (1/4, 1)$ and $C = \max_{i=1\ldots n}\|\mathbf{b}_i\|^2$. Then Algorithm 4 performs $\mathcal{O}\left(n^3 \log_{1/\delta}(C)\right)$ iterations of the `while-loop` in line 2 and a total of*

$$\mathcal{O}\left(mn^4 \log_{1/\delta}(C)\right)$$

arithmetic operations.

2.4.2 Experimental results

Extensive experiments have been made to examine how the classical LLL reduction algorithm performs in practice [NS06, GN08]. We ran some experiments to compare our PotLLL algorithm to our implementations[1] of LLL, DeepLLL, and BKZ.

Setting

We run the following algorithms, each with the standard reduction parameter $\delta = 0.99$:

1. classical LLL,
2. PotLLL,
3. DeepLLL with blocksize $\beta = 5$ and $\beta = 10$ (the latter up to dimension 240 only),
4. BKZ with blocksize 5 (BKZ-5) and 10 (BKZ-10).

The implementations all use the same arithmetic back-end. Integer arithmetic is done using GMP, and Gram-Schmidt arithmetic is done as described in [NS05, Figures 4 and 5]. As floating point types, `long double` (x64 extended precision format, 80 bit representation) and MPFR arbitrary precision floating point numbers are used with a precision as described in [NS05]. The implementations of DeepLLL and BKZ follow the classical description in [SE94]. PotLLL was implemented as described in Algorithm 4 (page 19).

We ran experiments in dimensions 40 to 300, considering the dimensions which are multiples of 10 from 40 to 300. In each dimension, we considered 50 random lattices in the sense of Goldstein and Mayer [GM03]. More precisely, we used the lattices of seed 0 to 49 from the SVP Challenge[2]. For each lattice, we used two bases: the original basis and a 0.75-LLL reduced basis.

All experiments were run on Intel® Xeon® X7550 CPUs at 2 GHz on a shared memory machine. For dimensions 40 up to 160, we used `long double` arithmetic, and for dimensions 160 up to 300, we used MPFR. In dimension 160, we did the experiments both using `long double` and MPFR[3] arithmetic. The reduced lattices did not differ. In dimension 170, floating point errors prevented the `long double` arithmetic variant to complete on some of the lattices.

[1] Implementations by Felix Fontein

[2] `http://www.latticechallenge.org/svp-challenge`

[3] `http://www.mpfr.org/`

Results

One common way to measure the quality of a reduction algorithm is by means of the Hermite factor $\frac{\|\mathbf{b}_1\|}{\mathrm{vol}(\mathcal{L})^{1/n}}$ to be achieved. There have been indications that modern lattice basis reduction algorithms such as LLL and BKZ achieve a Hermite factor which is exponential in n, more concretely that they achieve a Hermite factor of the form c^n, where c is a constant depending on the reduction algorithm only (cf. Section 2.2 and [NS06, GN08]). This is why for each dimension and each reduction algorithm we compute the average of the n-th root of the Hermite factor, i.e. $\frac{\|\mathbf{b}_1\|^{1/n}}{\mathrm{vol}(\mathcal{L})^{1/n^2}}$. An overview on the average values in some dimensions can be seen in Table 1. Note that our data for LLL is comparable to the one in [NS06] and [GN08, Table 1].

Dimension	$n = 100$	$n = 160$	$n = 220$	$n = 300$
Worst-case bound (proven)	≈ 1.0774	≈ 1.0777	≈ 1.0778	≈ 1.0779
Empirical 0.99-LLL	1.0187	1.0201	1.0207	1.0212
Empirical 0.99-BKZ-5	1.0154	1.0158	1.0161	1.0163
Empirical 0.99-PotLLL	1.0146	1.0150	1.0152	1.0153
Empirical 0.99-DeepLLL with $\beta = 5$	1.0138	1.0142	1.0147	1.0150
Empirical 0.99-BKZ-10	1.0140	1.0143	1.0144	1.0145
Empirical 0.99-DeepLLL with $\beta = 10$	1.0128	1.0132	1.0135	—

Table 1: Worst case bound and average case estimate for δ-LLL, δ-DeepLLL, δ-PotLLL and δ-BKZ reduction of the n-th root Hermite factor $\|\mathbf{b}_1\|^{1/n} \cdot \mathrm{vol}(\mathcal{L})^{-1/n^2}$. The entries are sorted in descending order with respect to the observed Hermite factors.

Further we are interested in the running time of the respective algorithms. Again for each dimension and each reduction algorithm the average of logarithm of the CPU time consumed is computed. The results are summarized in Figures 1, 2 and 3.

Figure 1 Figure 1a shows the average of the n-th root of the Hermite factor achieved by the different reduction algorithms in the different dimensions. We see a clear hierarchy. LLL performs by far the worst. Then comes BKZ-5, PotLLL, DeepLLL-5 and BKZ-10 and DeepLLL-10 performs best. DeepLLL-5 outperforms BKZ-10 in low dimensions ($n < 140$), for higher dimensions the order changes however. Interestingly BKZ with blocksize $\beta = 5$ is outperformed by DeepLLL with the same blocksize. Our new PotLLL algorithm clearly outperforms LLL and seems to be between BKZ-5 and BKZ-10. For dimension ≤ 300 it is worse than DeepLLL-5, however the graph indicates that this hierarchy could change in higher dimension.

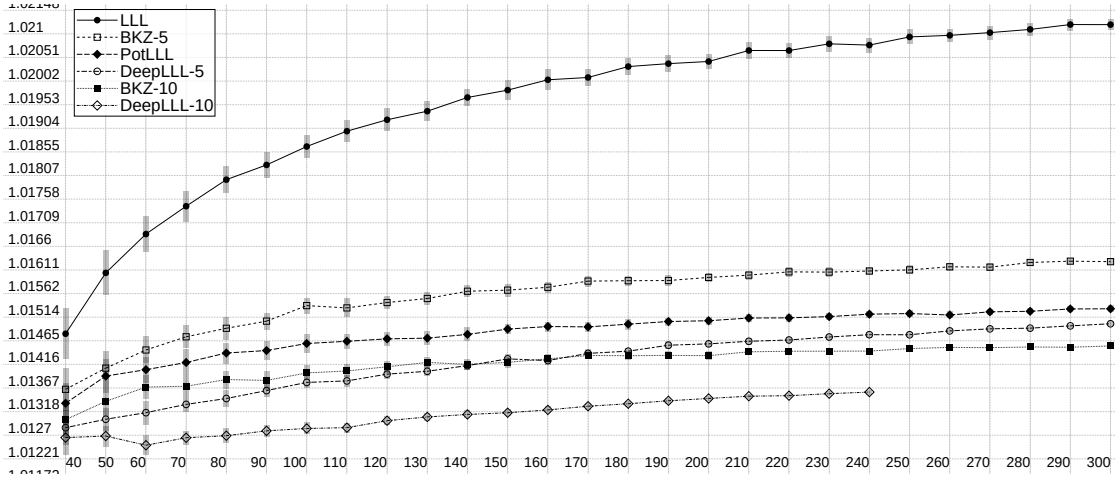
When it comes to the running time (see Figure 1b), we see that LLL (worst when it comes to the achieved Hermite factor) is the fastest, and DeepLLL with blocksize 10

(best when it comes to the achieved Hermite factor) is slowest. Further it is interesting to see that the running time of PotLLL and BKZ-5 is comparable. Note that the 'jump' in the running time at dimension 160 comes from the fact that in high dimensions the long double versions of the algorithms run into problems, this is why the MPFR arithmetic is used here. The use of MPFR in high dimensions does not change the hierarchy however.

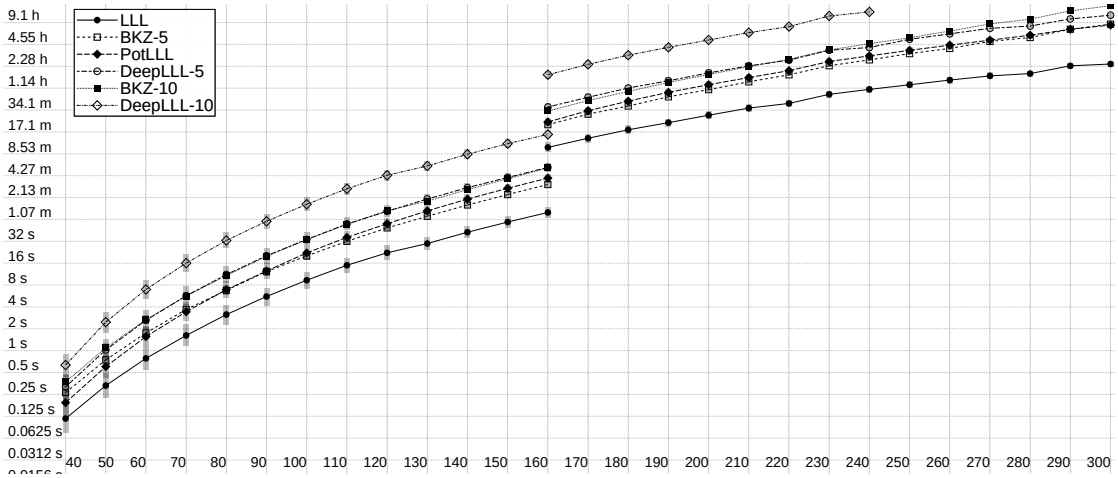
Figure 2 Figure 2 shows the behaviour of the `long double` versions of the algorithms. Every line connecting bullets corresponds to the behaviour of one algorithm for different dimensions, namely in dimensions $n = 40, 80, 120, 160$. In Figure 2a we see the behaviour of the algorithms applied to the original bases in HNF form. In Figure 2a we see the behaviour of the algorithms applied to the preprocessed bases, i.e. bases that were run through 0.75-LLL before.

The lower the bullet, the better the Hermite factor achieved. Similarly the more left the bullet, the faster the algorithm runs. As a consequence if the bullet of one algorithm is lower and more to the left than the bullet of another algorithm in the same dimension, there is no reason to use the second algorithm. It is interesting to see that while the hierarchy when it comes to the Hermite factor is independent on whether the bases were preprocessed or not, the hierarchy when it comes to the running time can change. This can be seen especially when comparing PotLLL with the BKZ variants.

Figure 3 The same comparisons as in the last Paragraph are made when the MPFR versions of the algorithms are used. Again we see that PotLLL seems to outperform BKZ with blocksize 5 when applied to preprocessed bases in high dimensions.

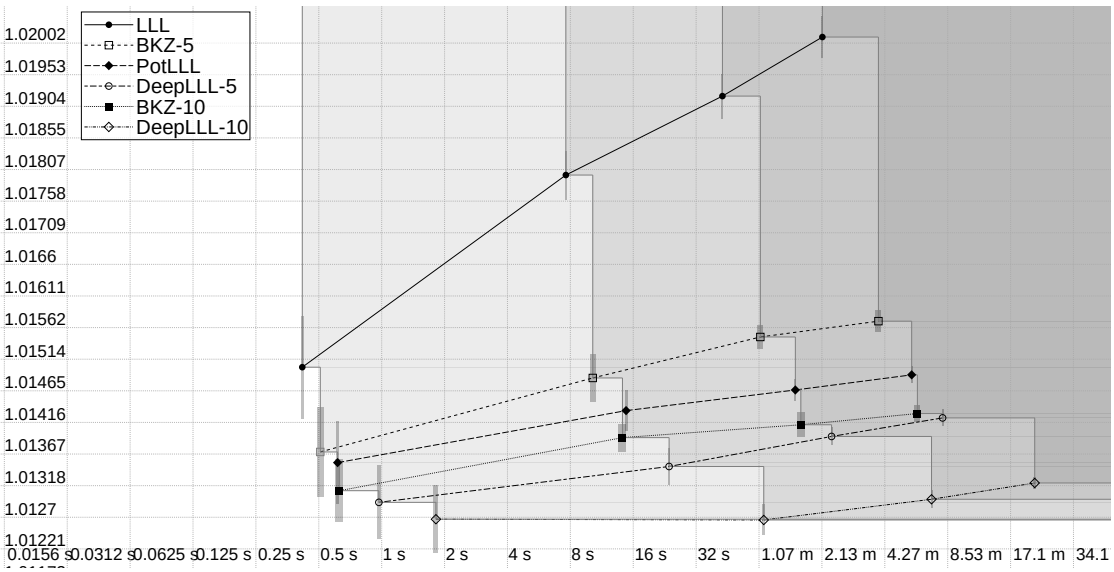


(a) Average n-th root Hermite factor.

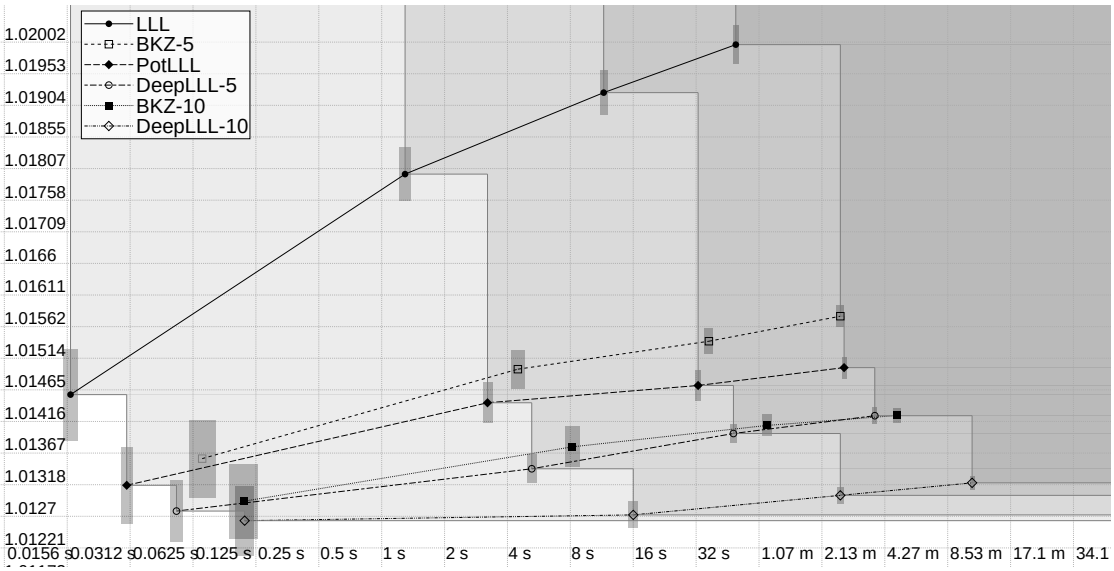


(b) Average logarithmic CPU time.

Figure 1: Overview of performance of the algorithms for dimensions n (x axis) from 40 to 300 (using MPFR for $n \geq 160$).

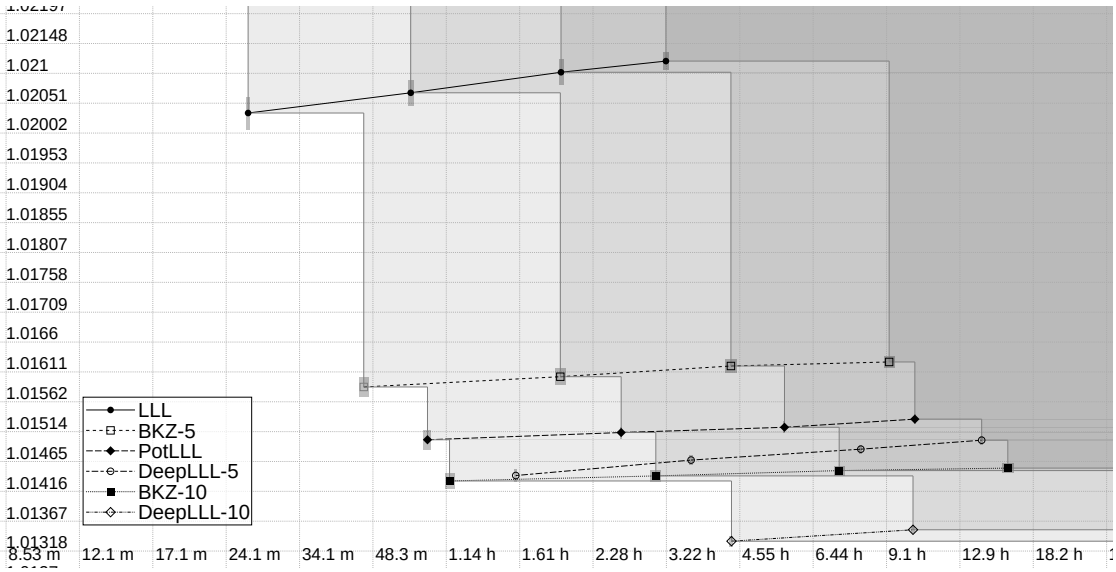


(a) Performance of the algorithms on the original bases.

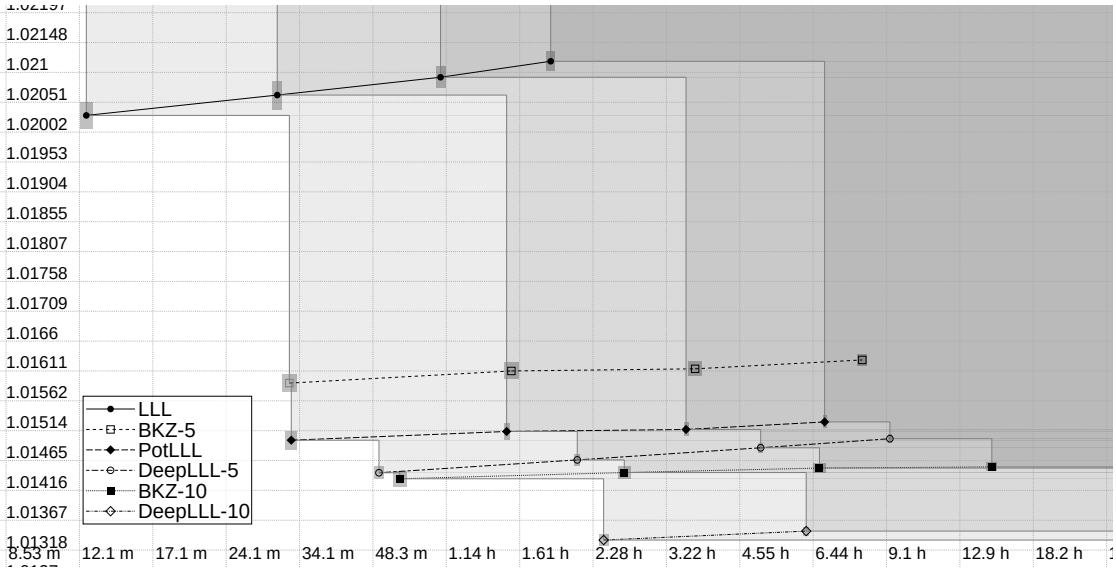


(b) Performance of the algorithms on the preprocessed bases (0.75-LLL reduced bases).

Figure 2: Comparison of n-th root Hermite factor (y axis) vs. running times (x axis) using `long double` arithmetic. The highlighted areas represent dimensions 40, 80, 120 and 160.



(a) Performance of the algorithms on the original bases.



(b) Performance of the algorithms on the preprocessed bases (0.75-LLL reduced bases).

Figure 3: Comparison of n-th root Hermite factor (y axis) vs. running times (x axis) using MPFR arithmetic. The highlighted areas represent dimensions 180, 220, 260 and 300.

2.5 Possible extensions

In this section we discuss some possible extensions of the work done. The ideas presented here are still in an early stage and could serve for future projects.

2.5.1 Weighted potential

In Algorithm 3 we proposed an algorithm to compute a δ-PotLLL reduced basis. In each iteration of the `while-loop` (line 2) we look for the insertion which minimizes the potential of the basis (line 4). If this insertion reduces the potential $\mathrm{Pot}(B) = \prod_i \det(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i))^2$ by a factor at least δ, the insertion is done.

Suppose for example that for $1 \leq k < l \leq n$ we have a permutation of the basis vectors which reduces $\det(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_k))$ by a factor $c < 1$ and increases $\det(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_l))$ by c^{-1} leaving $\det(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)), i \neq k, l$ unchanged. Intuitively we expect this to be an insertion improving the quality of the basis, while PotLLL will not perform this insertion. This leads to the following idea: Instead of considering the potential as in Definition 2.1.1 one could consider a weighted potential giving the determinants of the low rank sublattices of the form $\mathbf{b}_1, \ldots, \mathbf{b}_i$ more weight. For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Q}^n$ let the *weighted potential* be

$$\mathrm{Pot}_\alpha(B) := \prod_{i=1}^{n} \det\left(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)\right)^{2\alpha_i}.$$

Choosing α such that $\alpha_1 > \alpha_2 > \cdots > \alpha_n$, insertions at positions with low indices have priority then. It would be interesting to see how different choices for the α's influence the behaviour of the algorithm.

2.5.2 Simulated annealing

The PotLLL algorithm can be seen as attempt to minimize the potential of a basis. The algorithm stops when the potential of the basis can not be reduced by a factor at least δ by a deep insertion as defined in (2.1.1). For δ close to 1 this can be seen as a local minimum and it is possible that after an insertion that increases the potential, another local minimum can reached by again applying potential reducing insertions.

Simulated annealing is a well-known metaheuristic to avoid getting stuck in local minimums too quickly (e.g. [OG89]). In Algorithm 5 we propose how the simulated annealing principle could be applied to PotLLL. We need some preparations. Let $(T_t)_{t\in\mathbb{N}}$ be a sequence with

$$T_1 \geq T_2 \geq \ldots \quad \text{and} \quad \lim_{t\to\infty} T_t = 0, \tag{2.5.7}$$

and $P_\delta : \mathbb{R}^2 \to [0, 1]$ be a function with the following properties:

$$\begin{array}{lrl} \text{for} & T > T', x > \delta: & P_\delta(x, T) > P_\delta(x, T'), \\ \text{for} & x > \delta: & \lim_{T\to 0} P_\delta(x, T) = 0, \\ \text{for} & x < \delta: & P_\delta(x, T) = 1. \end{array} \tag{2.5.8}$$

Algorithm 5: PotLLL with simulated annealing

Input: Basis $B \in \mathbb{Z}^{n\times m}$, $\delta \in (1/4, 1]$, $(T_t)_{t\in\mathbb{N}}$ as in (2.5.7), P_δ as in (2.5.8)
Output: A δ-PotLLL reduced basis.

```
1  l ← 2
2  t ← 0
3  while l ≤ n do
4  |  t ← t + 1
5  |  Size-reduce(B)
6  |  k ← argmin_{1≤j≤l} Pot(σ_{j,l}B)
7  |  x ← Pot(σ_{k,l}B)/Pot(B)
8  |  p ← P_δ(x, T_t)
9  |  r ← Random(0, 1) # Random number in [0, 1]
10 |  if p < r then
11 |  |  B ← σ_{k,l}B
12 |  |  l ← k
13 |  else
14 |  |  l ← l + 1
15 |  end
16 end
17 return B
```

Consider again Algorithm 5. We see that in the `while-loop` starting on line 3 whenever $\mathrm{Pot}(\sigma_{k,l}B) < \delta\mathrm{Pot}(B)$ the deep insertion is done. In the case where $\mathrm{Pot}(\sigma_{k,l}B) \geq \delta\mathrm{Pot}(B)$ the deep insertion is done with a probability depending on the temperature T_t and the ratio of $\mathrm{Pot}(\sigma_{k,l}B)$ and $\mathrm{Pot}(B)$. The higher the temperature and the lower the ratio the higher the probability that the insertion is done. If the temperature is close to zero, the algorithm behaves like the original PotLLL algorithm, ensuring that the output is in fact a δ-PotLLL reduced basis.

2.5.3 PotBKZ

The BKZ reduction algorithm is currently the most practical algorithm for strong lattice reduction. It relies on a subroutine (subBKZ) computing the shortest vector within a projected sublattice (see Algorithm 6 for input/output behaviour).

Algorithm 6: subBKZ

Input: Basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, $1 \leq k < l \leq n$.
Output: $B' = [\mathbf{b}'_1, \ldots, \mathbf{b}'_n]$ with $\mathcal{L}(B) = \mathcal{L}(B')$, $\mathbf{b}_i = \mathbf{b}'_i$ for $i \in \{1, \ldots, k-1, l+1, \ldots, n\}$ and $\|\pi_k(\mathbf{b}_k)\| = \lambda_1\big(\pi_k\big(\mathcal{L}(\mathbf{b}'_k, \ldots, \mathbf{b}'_l)\big)\big)$.

As such the running time of the BKZ algorithm is lower bounded by the complexity of the subroutine, which is exponential in blocksize $\beta = l - k + 1$. However no useful upper

bound on the number of calls to this subroutine is known [HPS11] and a commonly used way around is to early abort the BKZ algorithm. Hanrot et al. [HPS11] showed that if the subroutine in fact computes a HKZ-basis of the projected basis $\pi_k(\mathcal{L}(\mathbf{b}_k, \ldots, \mathbf{b}_l))$, then terminating BKZ after

$$\Omega\left(\frac{n^3}{\beta^2}(\log n + \log\log \max_i \|\mathbf{b}_i\|)\right),$$

calls to the subroutine, the first basis vector of the basis returned by the algorithm has norm $\leq 2\nu_\beta^{\frac{n-1}{2(\beta-1)}+\frac{3}{2}} \det(\mathcal{L})^{1/n}$ where ν_β is the maximum of the Hermite's constants in dimensions $\leq \beta$.

We think that it might be interesting to examine the following early abortion strategy: The algorithm stops if no call to the subroutine (with the given blocksize) further reduces the potential of the basis (see Algorithm 7 for a corresponding adaption of the BKZ algorithm proposed in [SE94]). We call such a basis with this output property *PotBKZ reduced*:

Definition 2.5.1 *A basis B is called* δ-PotBKZ reduced *with respect to the subroutine sub-BKZ and blocksize β if*

1. $\forall 1 \leq j < i \leq n : |\mu_{i,j}| \leq \frac{1}{2}$.
2. $\forall 1 \leq k < n : \delta \cdot \mathrm{Pot}(B) \leq \mathrm{Pot}\big(subBKZ\big(B, k, \min\{k+\beta-1, n\}\big)\big)$.

It is not hard to see that as in the PotLLL algorithm, the PotBKZ algorithm terminates after a polynomial number of iterations in the main loop. It would be interesting to see how the algorithm performs with different variants of the subroutine.

Algorithm 7: PotBKZ

Input: Basis $B \in \mathbb{Z}^{n \times m}$, $\delta \in (1/4, 1]$, $2 \leq \beta \leq n$. Subroutine sub-BKZ

Output: A δ-PotBKZ reduced basis with respect to sub-BKZ and blocksize β.

δ-LLL-reduce(B)
$l \leftarrow 0$
$z \leftarrow 0$
while $z < n - 1$ **do**
  $l \leftarrow l + 1$
  $k \leftarrow \min\{l + \beta - 1, n\}$
  if $j = n$ **then**
    $j \leftarrow 1$
    $k \leftarrow \beta$
  end
  $B' \leftarrow$ sub-BKZ(B, k, l)
  if Pot(B') $< \delta$(Pot(B)) **then**
    $B \leftarrow B'$
    $z \leftarrow 0$
  else
    $z \leftarrow z + 1$
  end
  Size-reduce(B)
end
return B

Chapter 3

Closest point search based on dual HKZ bases

In this chapter we present the work done on the closest point search based on dual HKZ bases leading to [WM12].

Given an arbitrary point $\mathbf{t} \in \mathbb{R}^m$ the search version of the closest vector problem (CVP) asks for a closest lattice point of a given lattice $\mathcal{L} \subset \mathbb{R}^m$, i.e. for $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{w} - \mathbf{t}\|$ for all $\mathbf{w} \in \mathcal{L}$. This is equivalent to finding the closest lattice point to the orthogonal projection of $\mathbf{t}$ onto $\text{span}(\mathcal{L})$ and we will therefore restrict ourselves to full rank lattices in $\mathbb{R}^n$ in this chapter. While the problem is proven to be NP-hard (see e.g. [MG02]), algorithms exist to solve the problem approximately in polynomial time. Babai's nearest plane algorithm [Bab86] is the generic way to get an approximate solution, and the quality of the solution substantially depends on the quality of the basis it is applied to. The algorithm recursively selects the nearest $n-1, n-2, \ldots, 0$ dimensional plane spanned by the basis vectors to find a close vector. The more orthogonal the basis vectors are, the better the output of the algorithm is. E.g. if the basis is LLL-reduced, it finds a point lying within $2(4/3)^{n/2}$ times the distance of a closest lattice point to $\mathbf{t}$ [MG02]. In the extreme case where the basis vectors are pairwise orthogonal (note that such a basis does not necessarily exist), it even returns a closest vector. Babai's nearest plane algorithm can be modified to output an exact solution by not only considering the nearest, but all planes with distance up to a certain bound in the recursion steps. This is exactly the approach of Kannan [Kan83]. Once a plane is fixed, the problem translates to finding a closest lattice point in a lower dimensional lattice, namely the orthogonal projection of the lattice onto that plane. The number of planes to be considered in the lower dimensional lattice is dependent on the choice of the plane in the upper dimension which was realized by Fincke and Pohst [FP85]. So instead of searching all points inside a parallelotope, all points inside a hyperellipsoid are considered. The running time of Kannan's and Pohst's approach was proven to be $\mathcal{O}(n^n)$ in [Kan87]. Recently, refined analysis by Hanrot and Stehlé showed [HS10] that applying Kannan's algorithm to an HKZ-basis the closest vectors can be found by enumerating $2^{\mathcal{O}(n)} n^{0.5n}$ points. A more elaborate survey on different methods to solve the problem exactly can be found in e.g.

[AEVZ02].

In [Blö00], a different approach than the one of Kannan [Kan87] is presented. The main difference is that the basis used for closest point search is dual HKZ-reduced, e.g. it is a basis whose dual is HKZ-reduced. Due to the special form of the basis, the Transference Theorems (cf. Theorem 3.2.5), relating the lengths of the consecutive minimas and the covering radius of a lattice and its dual lattice, can be used to bound the number of planes to be considered. In each recursion step the number of planes to be considered decreases by 1. Having n planes to consider in the first recursion step, this results in enumeration of $n!$ lattice points.

In this chapter we give a refined analysis of the approach given in [Blö00]. We show how the overall expected number of points to be enumerated can be decreased. The adjunct 'expected' indicates that the Gaussian Heuristic is used to approximate the number of lattice points in a given subset of $\mathbb{R}^n$. While in the original algorithm the number of choices of the planes is bounded independently in each step, we examine how the choice of a plane in early recursion steps influences the possible number of choices in following steps. We show how to decrease the expected number of lattice points to be enumerated by an exponential factor $(\pi/4)^{n/2}$ by deriving how the choices of the planes in two consecutive recursion steps are connected. Further we derive a recursive formula (in the dimension of the lattice) for the expected number of points to be enumerated when the choices made in early recursion steps are rigorously used to constrain the further choices. A closed form approximation of this formula is still an open problem. However numerical computations show that this number can be bounded by $n^{0.75n}$ for $10 < n \leq 2000$.

The rest of the chapter is organized as follows. In Section 3.1 we discuss the principle of enumeration in a general setting. Then the notion of a dual lattice is introduced in Section 3.2. In Section 3.3 the original algorithm to find the closest vectors based on dual HKZ bases as proposed in [Blö00] is described. In Section 3.4 we show how the expected number of points to be enumerated can be decreased by a factor $(\pi/4)^{n/2}$. In Section 3.5, a recursive formula bounding the expected number of points to be enumerated is derived and its behavior is analyzed. Some concluding remarks are given in Section 3.6.

3.1 Enumeration

Both the shortest vector problem and the closest vector problem can be solved exactly using so called enumeration algorithms [FP85, SE94]. In this section we give a general description of the enumeration principle. This will ease the discussion in the rest of the chapter, where a special instance of the enumeration algorithm is described.

Let B be a basis of a lattice $\mathcal{L} \subset \mathbb{R}^m$. For a lattice vector $\mathbf{v} \in \mathcal{L}$, let $(v_1, \ldots, v_n) \in \mathbb{Z}^n$ be its coordinates with respect to B, i.e. $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n$. Given a sequence of sets $\mathbf{V}_i \subset \mathbb{Z}^{n-i+1}$, enumeration algorithms enumerate all $(v_1, \ldots, v_n) \in \mathbb{Z}^n$ satisfying that $(v_i, \ldots, v_n) \in \mathbf{V}_i$ for all $i = 1, \ldots, n$ in a recursive way from $i = n, \ldots, 1$: For given $(v_{i+1}, \ldots, v_n)$ enumerate all v_i such that $(v_i, \ldots, v_n) \in \mathbf{V}_i$. The number N of tuples

considered is hence upper bounded by

$$N \leq \sum_{i=1}^{n} |\mathbf{V}_i| = \mathcal{O}\left(\max_{i=1,\ldots,n} |\mathbf{V}_i|\right).$$

Let us now see how the sets $\mathbf{V}_i$ are defined in practice. Suppose we would like to find a lattice vector $\mathbf{v}$ closest to some given point $\mathbf{t} = t_1\mathbf{b}_1 + \cdots + t_n\mathbf{b}_n \in \operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, $(t_1, \ldots, t_n) \in \mathbb{R}^n$. Let B^* as usual denote the Gram-Schmidt orthogonalized basis of B,

$$B^T = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ \mu_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mu_{n,1} & \ldots & \mu_{n,n-1} & 1 \end{pmatrix} B^{*T}.$$

For a lattice point $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n \in \mathcal{L}$ we have

$$\mathbf{t} - \mathbf{v} = \sum_{j=1}^{n}(t_j - v_j)\mathbf{b}_j = \sum_{j=1}^{n} \underbrace{\left(t_j - v_j + \sum_{i=j+1}^{n}(t_i - v_i)\mu_{ij}\right)}_{=:e_j} \mathbf{b}_j^* = \sum_{j=1}^{n} e_j\mathbf{b}_j^*. \tag{3.1.1}$$

It is important to note that e_j depends on $v_j, \ldots, v_n$ and $t_j, \ldots, t_n$ only. We have $\|\mathbf{t} - \mathbf{v}\|^2 = \sum_{j=1}^{n} e_j^2\|\mathbf{b}_j^*\|^2$ and it is somewhat natural to define the set $\mathbf{V}_i$ by means of some restrictions on the e_i's. These restrictions are usually of the following form.

Let there be a division of $[\![1, n]\!] = [1, n] \cap \mathbb{Z}$ into k intervals as follows

$$[\![1, n]\!] = [\![\alpha_1, \beta_1]\!] \sqcup [\![\alpha_2, \beta_2]\!] \sqcup \cdots \sqcup [\![\alpha_k, \beta_k]\!],$$

where $\sqcup$ denotes the disjoint union. Further for $1 \leq i \leq n$ let $\alpha(i)$ denote the start point, $\beta(i)$ denote the end point of the interval i lies in, i.e.

$$\alpha(i) := \max\{\alpha_j : \alpha_j \leq i\} \quad \text{and} \quad \beta(i) := \min\{\alpha_j : \beta_j \geq i\}.$$

For $r_j \in \mathbb{R}_{>0}$, $j = 1, \ldots, n$ define

$$\mathbf{E}_i := \left\{ (x_i, \ldots, x_n) \in \mathbb{R}^{n-i+1} : \sum_{j=i}^{\beta(i)} \left(\frac{x_j}{r_j}\right)^2 \leq 1 \text{ and } (x_{i+1}, \ldots, x_n) \in \mathbf{E}_{i+1} \right\}$$

and with $(e_i, \ldots, e_n)$ as in (3.1.1) we define

$$\mathbf{V}_i := \left\{ (v_i, \ldots, v_n) \in \mathbb{Z}^{n-i+1} : (e_i, \ldots, e_n) \in \mathbf{E}_i \right\}.$$

We will now show by induction on i that for given $v_{i+1}, \ldots, v_n$ it is easy to enumerate all $v_i \in \mathbb{Z}$ such that $v_i, \ldots, v_n \in \mathbf{V}_i$. Clearly for $i = n$ we have

$$v_n \in \mathbf{V}_n \Leftrightarrow \left(\frac{e_n}{r_n}\right)^2 \leq 1 \Leftrightarrow (v_n - t_n)^2 \leq r_n^2 \Leftrightarrow v_n \in \underbrace{[\![-r_n + t_n, r_n + t_n]\!]}_{=[\![-r_n, r_n]\!] + t_n}.$$

Assume now that $v_{i+1}, \ldots, v_n$ are given satisfying $v_i, \ldots, v_n \in \mathbf{V}_i$. Under this condition we have the following equivalences:

$$\begin{aligned}
& (v_i, \ldots, v_n) \in \mathbf{V}_i \\
\Leftrightarrow \quad & \sum_{j=i}^{\beta(i)} \left(\frac{e_j}{r_j}\right)^2 \leq 1 \Leftrightarrow \left(\frac{e_i}{r_i}\right)^2 \leq 1 - \sum_{j=i+1}^{\beta(i)} \left(\frac{e_j}{r_j}\right)^2 \\
\Leftrightarrow \quad & \left(v_i - t_i + \sum_{j=i+1}^{n} (v_j - t_j)\mu_{ij}\right)^2 \leq r_i^2 \left(1 - \sum_{j=i+1}^{\beta(i)} \left(\frac{e_j}{r_j}\right)^2\right) \\
\Leftrightarrow \quad & v_i \in \left[\!\!\left[-r_i \sqrt{1 - \sum_{j=i+1}^{\beta(i)} \left(\frac{e_j}{r_j}\right)^2}, r_i \sqrt{1 - \sum_{j=i+1}^{\beta(i)} \left(\frac{e_j}{r_j}\right)^2} \right]\!\!\right] + t_i - \sum_{j=i+1}^{n} (v_j - t_j)\mu_{ij}.
\end{aligned}$$

So the values v_i to be considered are simply the integers inside some interval that can be computed depending on $v_{i+1,} \ldots, v_n$. The rest of this section is devoted to computing the expected size of the sets $\mathbf{V}_i$. We will approximate the cardinality of $\mathbf{V}_i$ by the $n-i+1$ dimensional volume of $\mathbf{E}_i \subset \mathbb{R}^{n-i+1}$. Note that

$$(v_i, \ldots, v_n) \in \mathbf{V}_i \Leftrightarrow (e_i, \ldots, e_n) \in \mathbf{E}_i.$$

Further with

$$M_i := \begin{pmatrix} 1 & 0 & \ldots & 0 \\ \mu_{i+1,i} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mu_{n,i} & \ldots & \mu_{n,n-1} & 1 \end{pmatrix}$$

we have that

$$(e_i, \ldots, e_n) = (v_i, \ldots, v_n)M_i - (t_i, \ldots, t_n)M_i.$$

So we have that $(v_i, \ldots, v_n)$ is in $\mathbf{V}_i$ if and only if the corresponding lattice point $(v_i, \ldots, v_n)M_i \in \mathcal{L}(M_i)$ is in a translated copy of $\mathbf{E}_i$, i.e.

$$(v_i, \ldots, v_n) \in \mathbf{V}_i \Leftrightarrow (v_i, \ldots, v_n)M_i \in (t_i, \ldots, t_n)M_i + \mathbf{E}_i.$$

As a consequence, the cardinality $|\mathbf{V}_i|$ of $\mathbf{V}_i$ equals the number of lattice points of $\mathcal{L}(M_i)$ that lie in the translated set $(t_i, \ldots, t_n)M_i + \mathbf{E}_i$. By the Gaussian volume heuristic

$$|\mathbf{V}_i| \approx \frac{\mathrm{vol}(\mathbf{E}_i)}{\det M_i} = \mathrm{vol}(\mathbf{E}_i). \tag{3.1.2}$$

Using the following lemma we can approximate the number of points in $\mathbf{V}_i$. It follows from the well-known formulas for the volume of multidimensional ellipsoids.

Lemma 3.1.1 *Using the notation as above then for $i \in [\![\alpha_l, \beta_l]\!]$, $1 \leq l \leq k$ we have that*

$$\mathrm{vol}(\mathbf{E}_i) = \pi^{(n-i+1)/2} \Gamma\left(\frac{\beta_l - i + 1}{2} + 1\right)^{-1} \prod_{j=l+1}^{k} \Gamma\left(\frac{\beta_j - \alpha_j + 1}{2} + 1\right)^{-1} \prod_{j=i}^{n} r_j.$$

3.2 Dual lattices

To every lattice $\mathcal{L} \subset \mathbb{R}^m$ there is an associated dual lattice $\mathcal{L}^\times \subset \mathbb{R}^m$. In this section we give the definition and some properties of a dual lattice.

Definition 3.2.1 *Given a lattice $\mathcal{L} \subset \mathbb{R}^m$ the* dual lattice $\mathcal{L}^\times \subset \mathbb{R}^m$ *is defined by*

$$\mathcal{L}^\times := \{\mathbf{v} \in \text{span}(\mathcal{L}) : \langle \mathbf{v}, \mathbf{w} \rangle \in \mathbb{Z} \quad \forall \mathbf{w} \in \mathcal{L}\}$$

For every basis B of $\mathcal{L}$ we define a unique reverse dual basis $B^\times$ such that $\mathcal{L}(B^\times) = \mathcal{L}^\times$.

Definition 3.2.2 *Given a basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ for a lattice $\mathcal{L} \subset \mathbb{R}^m$ of rank n, then $B^\times = [\mathbf{b}_1^\times, \ldots, \mathbf{b}_n^\times]$ is called the* reverse dual basis *of B if and only if*

$$\mathbf{b}_i^\times \in \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \quad \textit{and} \quad \langle \mathbf{b}_i^\times, \mathbf{b}_j \rangle = \delta_{i,n-j+1},$$

where $\delta_{i,j}$ denotes the usual Kronecker delta.

The following remark is an immediate consequence from the definition of the reverse dual basis.

Remark 3.2.3 *For a given lattice vector $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n \in \mathcal{L}$, then*

$$v_i = \langle \mathbf{v}, \mathbf{b}_{n-i+1}^\times \rangle$$

is the i-th coordinate of v with respect to the basis B.

The following lemma shows that given the basis of a lattice $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ and its reverse dual basis $B^\times = [\mathbf{b}_1^\times, \ldots, \mathbf{b}_n^\times]$, then the dual of the sublattices $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-i})$, $0 \leq i < n$ is just the orthogonal projection of the dual lattice onto the orthogonal complement of the space spanned by the first i vectors of the reverse dual basis:

Lemma 3.2.4 *Let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ a basis with dual basis $B^\times = [\mathbf{b}_1^\times, \ldots, \mathbf{b}_n^\times]$. Then the reverse dual basis of $[\mathbf{b}_1, \ldots, \mathbf{b}_{n-j}]$ equals $[\pi_{j+1}(\mathbf{b}_{j+1}^\times), \ldots, \pi_{j+1}(\mathbf{b}_n^\times)]$, $n-1 \geq j \geq 0$.*

Proof: We start by showing that $\pi_{j+1}(\mathbf{b}_i^\times) \in \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-j})$ for $i \geq j+1$. Clearly $\pi_{j+1}(\mathbf{b}_i^\times) \in \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$. Also $\pi_{j+1}(\mathbf{b}_i^\times) \in \text{span}(\mathbf{b}_1^\times, \ldots, \mathbf{b}_j^\times)^\perp$. Hence

$$\pi_{j+1}(\mathbf{b}_i^\times) \in \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \text{span}(\mathbf{b}_1^\times, \ldots, \mathbf{b}_j^\times)^\perp = \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-j}).$$

It remains to show that $\langle \pi_{j+1}(\mathbf{b}_i^\times), \mathbf{b}_k \rangle = 1$ if $k = n+1-i$ and $\langle \pi_{j+1}(\mathbf{b}_i^\times), \mathbf{b}_k \rangle = 0$ if $k \in \{1, \ldots, n-j\} \backslash \{n+1-i\}$. This is straightforward as with $k < n-j+1$, it holds that

$$\langle \pi_{j+1}(\mathbf{b}_i^\times), \mathbf{b}_k \rangle = \langle \mathbf{b}_i^\times, \mathbf{b}_k \rangle.$$

□

The successive minimas and the covering radius of a lattice and its dual are related by the following theorem from the geometry of numbers by Banaszczyk [Ban93].

Theorem 3.2.5 (Transference Theorems) *The successive minimas $\lambda_i(\mathcal{L})$ and covering radius $\mu(\mathcal{L})$ of a lattice $\mathcal{L}$ of rank n satisfy the following bounds*

1. $\lambda_i(\mathcal{L}) \cdot \lambda_{n-i+1}(\mathcal{L}^\times) \leq n, \quad i = 1, \ldots, n,$
2. $\mu(\mathcal{L}) \cdot \lambda_1(\mathcal{L}^\times) \leq \frac{n}{2}.$

Let us now describe the closest point search approach of [Blö00] based on dual HKZ bases.

3.3 Original approach

In this section the closest point search approach of [Blö00] based on dual HKZ bases is described.

Definition 3.3.1 *A basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called HKZ-basis if and only if it satisfies the following two conditions*

1. $\forall 1 \leq j < i \leq n : \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \leq \frac{1}{2}$ *(size-reduced).*
2. *The i-th Gram-Schmidt vector satisfies $\|\mathbf{b}_i^*\| = \lambda_1(\pi_i(\mathcal{L}))$.*

Definition 3.3.2 *A basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called dual HKZ-basis when its reverse-dual basis $B^\times = [\mathbf{b}_1^\times, \ldots, \mathbf{b}_n^\times]$ is HKZ-reduced.*

The following lemma follows from Lemma 3.2.4:

Lemma 3.3.3 ([Blö00] Lemma 1) *If $[\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is a dual HKZ-basis for $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ then $[\mathbf{b}_1, \ldots, \mathbf{b}_k]$ is a dual HKZ-basis for $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$, $k = 1, \ldots, n$.*

Let as usual $B^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ denote the Gram-Schmidt orthogonalized basis of B. Clearly $\mathbf{b}_k^* \in \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$, and as $\langle \mathbf{b}_i, \mathbf{b}_k^* \rangle = 0$ for $i < k$ it follows that

$$\langle \mathbf{b}_k^*, \mathbf{b}_k \rangle = \langle \mathbf{b}_k^*, \mathbf{b}_k^* \rangle = \|\mathbf{b}_k^*\|^2.$$

So we have that $\frac{\mathbf{b}_k^*}{\|\mathbf{b}_k^*\|^2}$ is the first basis vector of the basis dual to $[\mathbf{b}_1, \ldots, \mathbf{b}_k]$. Hence we get the following corollary.

Corollary 3.3.4 *Using the notation from above, it holds that $\frac{\mathbf{b}_k^*}{\|\mathbf{b}_k^*\|^2}$ is a shortest vector in $\mathcal{L}^\times(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ and consequently $\frac{1}{\|\mathbf{b}_k^*\|} = \lambda_1(\mathcal{L}^\times(\mathbf{b}_1, \ldots, \mathbf{b}_k))$. Further in a dual HKZ reduced basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, $\|\mathbf{b}_k^*\|$ is maximal under all possible bases for the sublattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$.*

With Corollary 3.3.4 we have that $\mu(\mathcal{L}) \cdot \lambda_1(\mathcal{L}^\times) = \frac{\mu(\mathcal{L})}{\|\mathbf{b}_n^*\|}$, so the second inequality in the Transference Theorems (Theorem 3.2.5) implies that

$$\mu(\mathcal{L}) \leq \frac{n}{2}\|\mathbf{b}_n^*\|. \tag{3.3.3}$$

Le us now review how these considerations can be used to solve the search version of CVP. Given a lattice $\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ in $\mathbb{R}^n$ and a vector $\mathbf{t} \in \mathbb{R}^n$, we want to find a vector $\mathbf{v}$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{w} - \mathbf{t}\|$ for all $\mathbf{w} \in \mathcal{L}$. We assume that the basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is dual HKZ reduced. The following notation is used:

1. $\mathbf{e} = e_1\mathbf{b}_1^* + \cdots + e_n\mathbf{b}_n^* = \mathbf{t} - \mathbf{v}$ denotes the error vector,
2. $\mathbf{e}^{(i)} := \mathbf{e} - \sum_{j=i+1}^{n} e_j\mathbf{b}_j^*$ is the orthogonal projection of the error vector onto $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$,
3. $\mu^{(i)}$ denotes the covering radius of the sublattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$,

Suppose $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n$, $v_i \in \mathbb{Z}$ is a closest vector to $\mathbf{t} = t_1\mathbf{b}_1 + \cdots + t_n\mathbf{b}_n$, $t_i \in \mathbb{R}$. With (3.3.3) we get

$$\|\mathbf{v} - \mathbf{t}\| \leq \mu(\mathcal{L}) \leq \frac{n}{2}\|\mathbf{b}_n^*\|,$$

and as $(v_n - t_n)^2\|\mathbf{b}_n^*\|^2 \leq \|\mathbf{v} - \mathbf{t}\|^2 \leq \left(\frac{n}{2}\right)^2\|\mathbf{b}_n^*\|^2$ we have

$$|v_n - t_n| \leq \frac{n}{2}.$$

Hence we get an interval of length n for the n-th coordinate v_n of v:

$$v_n \in [t_n - n/2, t_n + n/2].$$

As $v_n \in \mathbb{Z}$, the expected number of values for v_n to enumerate equals the length of the interval, which is n.

Note that for the orthogonal projection $\mathbf{t}^{(n-1)}$ of $\mathbf{t} - v_n\mathbf{b}_n$ onto $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$ we have

$$\mathbf{t}^{(n-1)} = \mathbf{t} - v_n\mathbf{b}_n - \frac{\langle \mathbf{t} - v_n\mathbf{b}_n, \mathbf{b}_n^* \rangle}{\langle \mathbf{b}_n^*, \mathbf{b}_n^* \rangle}\mathbf{b}_n^* = \mathbf{t} - v_n\mathbf{b}_n - (t_n - v_n)\mathbf{b}_n^*, \tag{3.3.4}$$

and $(t_n - v_n) = e_n$. The following lemma allows to recursively carry the problem to proper sublattices of $\mathcal{L}$ in order to derive corresponding bounds for the other coordinates of $\mathbf{v}$.

Lemma 3.3.5 ([Blö00] Lemma 3) *A vector $\mathbf{w} \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ is a closest vector to $\mathbf{t} - \sum_{j>i} x_j\mathbf{b}_j$, $x_j \in \mathbb{Z}$ if and only if $\mathbf{w}$ is a closest vector of the orthogonal projection $\mathbf{t}^{(i)}$ of $\mathbf{t} - \sum_{j>i} x_j\mathbf{b}_j$ onto $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$.*

So given $v_{i+1}, \ldots, v_n$ and $e_{i+1}, \ldots, e_n$ the problem reduces to finding a closest vector to $\mathbf{t}^{(i)} = t - \sum_{j=i+1}^{n} v_j \mathbf{b}_j - \sum_{j=i+1}^{n} e_j \mathbf{b}_j^*$ in the lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ of rank i. As by Lemma 3.3.3, $[\mathbf{b}_1, \ldots, \mathbf{b}_i]$ is a dual HKZ basis for $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$, we can recursively carry the problem to a lower dimension. In dimension $i = 1$, $\mathbf{t}^{(1)} \in \operatorname{span}(\mathbf{b}_1)$ and we set $v_1 = \left\lfloor \frac{\langle \mathbf{t}^{(1)}, \mathbf{b}_1 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \right\rceil$ and $e_1 = \frac{\langle \mathbf{t}^{(1)}, \mathbf{b}_1 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} - \left\lfloor \frac{\langle \mathbf{t}^{(1)}, \mathbf{b}_1 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \right\rceil$ in order to get the closest lattice vector in $\mathcal{L}(\mathbf{b}_1)$ to $\mathbf{t}^{(1)}$. In fact

$$\mathbf{t}^{(1)} - v_1 \mathbf{b}_1 - e_1 \mathbf{b}_1^* = \mathbf{t} - \sum_{j=1}^{n} v_j \mathbf{b}_j - \sum_{j=1}^{n} e_j \mathbf{b}_j^* = 0,$$

assuring that we get a valid pair of vectors $\mathbf{v} \in \mathcal{L}$ and $\mathbf{e} \in \mathbb{R}^n$ in the sense that $\mathbf{v} + \mathbf{e} = \mathbf{t}$. Hence we have the following lemma:

Lemma 3.3.6 *Recursively we can derive $n!$ candidates for a closest vector to $\mathbf{t}$ in $\mathcal{L}$ given a dual HKZ-basis for $\mathcal{L}$.*

We will now give a short motivation for further analysis. The algorithm and the corresponding bound is not optimized at all. Suppose the n-th coordinate e_n of the error vector equals $\frac{n}{2}$. Clearly we have the following equality $\frac{n}{2} = \frac{\langle \mathbf{e}, \mathbf{b}_n^* \rangle}{\langle \mathbf{b}_n^*, \mathbf{b}_n^* \rangle} = \|\mathbf{e}\| \frac{1}{\|\mathbf{b}_n^*\|} \cos\gamma$, where γ is the angle between $\mathbf{b}_n^*$ and $\mathbf{e}$. As $\|\mathbf{e}\| \leq \mu(\mathcal{L})$, with Equation (3.3.3) we get $\frac{n}{2} \leq \frac{n}{2} \cos\gamma$. Consequently $\gamma = 0$ which means that the error vector points exactly in the direction of $\mathbf{b}_n^*$. So the error vector can be written as multiple of $\mathbf{b}_n^*$ and the coefficients $e_1, \ldots, e_{n-1}$ are trivially zero. In the next section we will see how the value of e_i influences the interval length in which e_{i-1} lies.

3.4 Local improvement

Using the notation of Section 3.1 we have seen that the original approach in Section 3.3 corresponds to enumeration as described in Section 3.1 using the error coefficient sets $\mathbf{E}_i^{\text{org}}$ which are recursively defined as follows

$$\mathbf{E}_i^{\text{org}} := \left\{ (x_i, \ldots, x_n) \in \mathbb{R}^{n-i+1} : x_i^2 \leq (i/2)^2 \text{ and } (x_{i+1}, \ldots, x_n) \in \mathbf{E}_{i+1}^{\text{org}} \right\}.$$

The expected number of points N to enumerate consequently is (cf. Equation (3.1.2))

$$N \approx \operatorname{vol}\left(\mathbf{E}_i^{\text{org}}\right) = n!.$$

The goal of this section is to define sets $\mathbf{E}_i$ such that still all closest vectors to $\mathbf{t}$ are in the set $\mathbf{V}_1 := \{(v_1, \ldots, v_n) \in \mathbb{Z}^n : (e_i, \ldots, e_n) \in \mathbf{E}_i\}$ but $\operatorname{vol}(\mathbf{E}_i) \leq \operatorname{vol}\left(\mathbf{E}_i^{\text{org}}\right)$, i.e. the expected number of elements to enumerate is reduced.

We will now show how additional constraints on the coefficients e_i of the error vector can be derived. Recall that the condition $|e_i| \leq \frac{i}{2}$ comes from the fact that $\|\mathbf{e}^{(i)}\| \leq \mu^{(i)} \leq \frac{i}{2} \|\mathbf{b}_i^*\|$, where the second inequality is due to the dual HKZ reducedness of the

basis. This implies $\|e_i\mathbf{b}_i^*\| \leq \|\mathbf{e}^{(i)}\| \leq \mu^{(i)} \leq \frac{i}{2}\|\mathbf{b}_i^*\|$ and consequently $|e_i| \leq \frac{i}{2}$. However $\|\mathbf{e}^{(i)}\| \leq \mu^{(i)}$ is not the only bound on $\|\mathbf{e}^{(i)}\|$ we have. So for $n \geq k > i \geq 1$ it holds that

$$\|\mathbf{e}^{(i)}\|^2 = \|\mathbf{e}^{(k)}\|^2 - \sum_{j=i+1}^{k} e_j^2\|\mathbf{b}_j^*\|^2 \leq \mu^{(k)2} - \sum_{j=i+1}^{k} e_j^2\|\mathbf{b}_j^*\|^2.$$

The following lemma shows that this bound is potentially smaller than $\mu^{(i)2}$.

Lemma 3.4.1 *Using the notation from above, for all $1 \leq i < k \leq n$ it holds that*

$$\mu^{(k)2} \leq \mu^{(i)2} + \frac{1}{4}\sum_{j=i+1}^{k} \|\mathbf{b}_j^*\|^2.$$

Proof: We show that for all $2 \leq k \leq n$, it holds that

$$\mu^{(k)2} \leq \mu^{(k-1)2} + \frac{1}{4}\|\mathbf{b}_k^*\|^2.$$

The claim follows inductively for $i \leq k-1$. It is enough to show that for $\mathbf{t} \in \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ with $\mathbf{t} = t_1\mathbf{b}_1 + \cdots + t_k\mathbf{b}_k$, there exist $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_k\mathbf{b}_k$ such that $\|\mathbf{v}-\mathbf{t}\|^2 \leq \mu^{(k-1)2} + \frac{1}{4}\|\mathbf{b}_k^*\|^2$. Choose $v_k = \lfloor t_k \rceil$. Then $\mathbf{t}^{(k-1)} = \mathbf{t} - v_k\mathbf{b}_k - (t_k - v_k)\mathbf{b}_k^* \in \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{k-1})$ (cf. Equation (3.3.4)) and there exist $v_1, \ldots, v_{k-1} \in \mathbb{Z}$ such that $\|v_1\mathbf{b}_1 + \cdots + v_{k-1}\mathbf{b}_{k-1} - \mathbf{t}^{(k-1)}\| \leq \mu^{(i)}$. So,

$$\begin{aligned} \|v_1\mathbf{b}_1 + \cdots + v_k\mathbf{b}_k - \mathbf{t}\|^2 &= \|v_1\mathbf{b}_1 + \cdots + v_{k-1}\mathbf{b}_{k-1} - (t_k - v_k)\mathbf{b}_k^* - \mathbf{t}^{(k-1)}\|^2 \\ &= \mu^{(i)2} + (t_k - v_k)^2\|\mathbf{b}_k^*\|^2 \\ &\leq \mu^{(i)2} + 1/4\|\mathbf{b}_k^*\|^2 \end{aligned}$$

□

So if $e_j^2 \geq \frac{1}{4}$, $j = i+1, \ldots, k$, we have a tighter upper bound

$$\|\mathbf{e}^{(i)}\|^2 \leq \mu^{(k)2} - \sum_{j=i+1}^{k} e_j^2\|\mathbf{b}_j^*\|^2 \leq \mu^{(i)2}. \tag{3.4.8}$$

Hence the size of the intervals in which the e_i's lie can be reduced. For all $i = 2, \ldots, n$, we derive factors $\alpha_i(e_i) \in \mathbb{R}$ depending on e_i, such that $\|\mathbf{e}^{(i-1)}\| \leq \alpha_i(e_i)\mu^{(i-1)}$ and consequently $|e_{i-1}| \leq \alpha_i(e_i)\frac{i-1}{2}$. For $x \in \mathbb{R}$ and $i \in \mathbb{N}$ let us define

$$\alpha_i^2(x) := \frac{\frac{i^2}{4} - x^2}{\frac{i^2}{4} - \frac{1}{4}}.$$

We obtain the following lemma:

Lemma 3.4.2 *If $x^2 \geq \frac{1}{4}$, then we have*

$$\mu^{(i)2} - x^2\|\mathbf{b}_i^*\|^2 \leq \alpha_i^2(x) \cdot \mu^{(i-1)2}.$$

Proof: We have to show that

$$\left(\frac{i^2}{4} - \frac{1}{4}\right)\left(\mu^{(i)2} - x^2\|\mathbf{b}_i^*\|^2\right) \leq \left(\frac{i^2}{4} - x^2\right)\mu^{(i-1)2}.$$

Since $\mu^{(i)2} - \frac{1}{4}\|\mathbf{b}_i^*\|^2 \leq \mu^{(i-1)2}$, it is sufficient to show that

$$\left(\frac{i^2}{4} - \frac{1}{4}\right)\left(\mu^{(i)2} - x^2\|\mathbf{b}_i^*\|^2\right) \leq \left(\frac{i^2}{4} - x^2\right)\left(\mu^{(i)2} - \frac{1}{4}\|\mathbf{b}_i^*\|^2\right).$$

This is true since

$$\left(x^2 - \frac{1}{4}\right)\mu^{(i)2} \leq \left(x^2 - \frac{1}{4}\right)\frac{i^2}{4}\|\mathbf{b}_i^*\|^2.$$

□

We can now prove the core lemma, which gives the factor by which the error vector is smaller than the covering radius.

Lemma 3.4.3 *Under the previous assumptions and notations:*

$$\|\mathbf{e}^{(i-1)}\|^2 \leq \alpha_i^2(e_i) \cdot \mu^{(i-1)2}.$$

Proof: We separate the two cases where $|e_i| < \frac{1}{2}$, $|e_i| \geq \frac{1}{2}$ respectively. If $|e_i| < \frac{1}{2}$, then $\alpha_i^2(e_i) > 1$ and the proposition follows by $\|\mathbf{e}^{(i-1)}\|^2 \leq \mu^{(i-1)2}$. If $|e_i| \geq \frac{1}{2}$, the claim follows from

$$\|\mathbf{e}^{(i-1)}\|^2 = \|\mathbf{e}^{(i)}\|^2 - e_i^2\|\mathbf{b}_i^*\|^2 \leq \mu^{(i)2} - e_i^2\|\mathbf{b}_i^*\|^2,$$

and Lemma 3.4.2. □

So with $e_{i-1}^2\|\mathbf{b}_{i-1}^*\|^2 \leq \|\mathbf{e}^{(i-1)}\|^2 \leq \alpha_i^2(e_i) \cdot \mu^{(i-1)2}$ and $\frac{\mu^{(i-1)2}}{\|\mathbf{b}_{i-1}^*\|^2} \leq \frac{(i-1)^2}{2}$ we immediately obtain the following bound.

Corollary 3.4.4 *Using the notation from before, for $i = 2, \ldots, n$, the coefficient pair (e_{i-1}, e_i) satisfies*

$$e_{i-1}^2\frac{i^2-1}{4} + e_i^2\frac{(i-1)^2}{4} \leq \frac{i^2}{4}\frac{(i-1)^2}{4}. \tag{3.4.9}$$

Recall that in the original approach (Section 3.3) we had the bound $|e_i| \leq i/2$ for $i = 1, \ldots, n$ giving an interval of length i to choose the integer coordinate v_i from. By Corollary 3.4.4 for $i = 1, \ldots, n-1$ these intervals might be substantially smaller dependent on the choice of v_{i+1} (which itself determines the value of e_{i+1}). Namely we have that

$$|e_i| \leq \sqrt{\frac{i^2}{(i+1)^2 - 1}\left(\frac{(i+1)^2}{4} - e_{i+1}^2\right)}.$$

Using this bound we can reduce the expected number of points to be enumerated. Recursively define

$$\mathbf{E}_i := \left\{(e_i, \ldots, e_n) \in \mathbb{R}^{n-i+1} : (e_i, e_{i+1}) \text{ satisfy Ineq. (3.4.9) and } (e_{i+1}, \ldots, e_n) \in \mathbf{E}_i\right\}.$$

Let us upper bound $\mathrm{vol}(\mathbf{E}_i)$. Assume first that $n - i + 1$ is even. Clearly

$$\mathbf{E}_i \subset \mathbf{E}'_i := \prod_{j=0}^{\frac{n-i-1}{2}} \left\{(e_{i+2j}, e_{i+2j+1}) \in \mathbb{R}^2 : \text{Inequality (3.4.9) holds } \right\}.$$

The volume of $\mathbf{E}'_i$ can be computed as the product of the volumes of 2-dimensional ellipses

$$\begin{aligned}
\mathrm{vol}\left(\mathbf{E}'_i\right) &= \prod_{j=0}^{\frac{n-i-1}{2}} \frac{\pi}{4}(i+2j+1)^2(i+2j)\frac{1}{\sqrt{(i+2j+1)^2-1}} \\
&= \left(\frac{\pi}{4}\right)^{(n-i+1)/2} \frac{n!}{(i-1)!} \prod_{j=0}^{\frac{n-i-1}{2}} \frac{i+2j+1}{\sqrt{(i+2j+1)^2-1}}.
\end{aligned}$$

As

$$\prod_{j=0}^{\frac{n-i-1}{2}} \frac{i+2j+1}{\sqrt{(i+2j+1)^2-1}} = \sqrt{\prod_{j=0}^{\frac{n-i-1}{2}} \frac{(i+2j+1)^2}{(i+2j)(i+2j+2)}} \leq \sqrt{2},$$

we get

$$\mathrm{vol}\left(\mathbf{E}'_i\right) \leq \sqrt{2}\left(\frac{\pi}{4}\right)^{(n-i+1)/2} \frac{n!}{(i-1)!}.$$

Similarly if $n - i + 1$ is odd, we have

$$\mathbf{E}_i \subset \mathbf{E}''_i := \{|x_1| \leq i/2\} \times \prod_{j=0}^{\frac{n-i-2}{2}} \left\{(x_{i+2j+1}, x_{i+2j+2}) \in \mathbb{R}^2 : \text{Equation (3.4.9) holds } \right\},$$

with

$$\begin{aligned}\mathrm{vol}(\mathbf{E}_i') &= i\prod_{j=0}^{\frac{n-i-2}{2}}\frac{\pi}{4}(i+2j+2)^2(i+2j+1)\frac{1}{\sqrt{(i+2j+2)^2-1}}\\ &= \left(\frac{\pi}{4}\right)^{(n-i)/2}\frac{n!}{(i-1)!}\prod_{j=0}^{\frac{n-i-2}{2}}\frac{i+2j+2}{\sqrt{(i+2j+2)^2-1}}\\ &\le \sqrt{\frac{3}{2}}\left(\frac{\pi}{4}\right)^{(n-i)/2}\frac{n!}{(i-1)!}.\end{aligned}$$

We immediately get that all closest vectors to a given point $\mathbf{t}\in\mathbb{R}^n$ can be found by recursively enumerating an expected number smaller than $\sqrt{2}\left(\frac{\pi}{4}\right)^{n/2}n!$ lattice points. So with $\sqrt{\pi/4}\approx 0.886$ we get an exponential gain of roughly 0.886^n compared to the original considerations.

3.5 Global improvement

Recall the starting point of the considerations of the previous section. We have an upper bound on $\|\mathbf{e}^{(i)}\|^2$, namely for all i,k with $1\le i\le k\le n$

$$\|\mathbf{e}^{(i)}\|^2 \le \mu^{(k)2} - \sum_{j=i+1}^{k} e_j^2\|\mathbf{b}_j^*\|^2. \tag{3.5.10}$$

Note that the bound in Inequality (3.5.10) is decreasing with increasing e_j's, $j=i+1,\ldots,k$, and in fact if they satisfy $|e_j|>\frac{1}{2}$ then as in Inequality (3.4.8), we have

$$\mu^{(k)2} - \sum_{j=i+1}^{k} e_j^2\|\mathbf{b}_j^*\|^2 < \mu^{(i)2}.$$

In the original approach (see Section 3.3), only the case $k=i$ was considered. In Section 3.4 we considered the case where $k=i+1$ and we got that

$$\|\mathbf{e}^{(i)}\|^2 \le \alpha_{i+1}^2(e_{i+1})\cdot\mu^{(i)2},$$

where $\alpha_{i+1}^2(e_{i+1}) := \left(\frac{(i+1)^2}{4}-e_{i+1}^2\right)\left(\frac{(i+1)^2}{4}-\frac{1}{4}\right)^{-1}$. From that we derived that pairs of coefficients (e_i,e_{i+1}) lie inside a 2-dimensional ellipsoid of volume $\frac{\pi}{4}(i+1)^2\sqrt{\frac{i}{i+2}}$. The goal of this section is to generalize this method to more than just 2-tuples of coefficients. Note that

$$\begin{aligned}\|\mathbf{e}^{(i-1)}\|^2 = \|\mathbf{e}^{(i)}\|^2 - e_i^2\|\mathbf{b}_i^*\|^2 &\le \alpha_{i+1}^2(e_{i+1})\cdot\mu^{(i)2} - e_i^2\|\mathbf{b}_i^*\|^2\\ &= \alpha_{i+1}^2(e_{i+1})\left(\mu^{(i)2} - \frac{e_i^2}{\alpha_{i+1}^2(e_{i+1})}\|\mathbf{b}_i^*\|^2\right).\end{aligned}$$

So under the condition that $\frac{e_i^2}{\alpha_{i+1}^2(e_{i+1})} \geq \frac{1}{4}$, by Lemma 3.4.2 we have that

$$\|\mathbf{e}^{(i-1)}\|^2 \leq \alpha_{i+1}^2(e_{i+1})\alpha_i^2\left(\frac{e_i^2}{\alpha_{i+1}^2(e_{i+1})}\right)\mu^{(i-1)2}.$$

If $|e_{i+1}|, |e_i| > \frac{1}{2}$, $\alpha_{i+1}^2(e_{i+1}) < 1$ and $\alpha_i^2\left(\frac{e_i^2}{\alpha_{i+1}^2(e_{i+1})}\right) < 1$. Clearly the bigger $|e_{i+1}|, |e_i|$ the smaller the bound on $\|\mathbf{e}^{(i-1)}\|$ becomes.

Definition 3.5.1 *For $e_n, \ldots, e_1$ recursively define $\beta_{n+1}^2, \ldots, \beta_1^2$ by*

$$\beta_{n+1}^2 := 1 \quad \textit{and} \quad \beta_{i-1}^2 := \begin{cases} 1 & \textit{if} \quad |e_{i-1}| < \frac{1}{2}, \\ \beta_i^2\alpha_{i-1}^2\left(\frac{e_{i-1}}{\beta_i}\right) & \textit{otherwise.} \end{cases}$$

Note that $\beta_i^2 \leq 1$ for all i.

Proposition 3.5.2 *For $i = n, \ldots, 2$ we have*

$$\|\mathbf{e}^{(i-1)}\|^2 \leq \beta_i^2\mu^{(i-1)2}.$$

Proof: The proof goes by reverse induction on i. For $i = n-1$ the result follows by Proposition 3.4.3. Assume the results holds for i. If $|e_i| < \frac{1}{2}$, then $\beta_i^2 = 1$ and the proposition follows trivially. For the case $|e_i| \geq \frac{1}{2}$, note that

$$\|\mathbf{e}^{(i-1)}\|^2 = \|\mathbf{e}^{(i)}\|^2 - e_i^2\|\mathbf{b}_i^*\|^2 \leq \beta_{i+1}^2\mu^{(i)2} - e_i^2\|\mathbf{b}_i^*\|^2 = \beta_{i+1}^2\mu^{(i)2} - e_i^2\|\mathbf{b}_i^*\|^2.$$

Also $\frac{e_i^2}{\beta_{i+1}^2} > \frac{1}{4}$ and with Lemma 3.4.2, we get that

$$\beta_{i+1}^2\left(\mu^{(i)2} - \frac{e_i^2}{\beta_{i+1}^2}\|\mathbf{b}_i^*\|^2\right) \leq \beta_{i+1}^2\alpha_i^2\left(\frac{e_i}{\beta_{i+1}}\right)\mu^{(i-1)2} = \beta_i^2\mu^{(i-1)2}.$$

□

By the Transference Theorems the following corollary follows immediately:

Corollary 3.5.3 *For $i = n, \ldots, 2$ we have*

$$e_{i-1}^2 \leq \beta_i^2 \cdot \left(\frac{i-1}{2}\right)^2.$$

Similar to Corollary 3.4.4 this gives an interval of length $|(i-1)\cdot\beta_i|$ for the possible choices of v_{i-1}. Under the assumption that a few consecutive e_j's are at least one half in absolute value, e.g. $|e_k|, \ldots, |e_i| \geq \frac{1}{2}$, the next lemma will give a closed form expression for β_i depending on $e_k, \ldots, e_i$. As a corollary of the next lemma and Proposition 3.5.2, we will see how $e_k, \ldots, e_i$ satisfy a $(k-i+1)$-dimensional ellipsoid equation.

Lemma 3.5.4 *Let $n \geq k \geq i \geq 1$. Under the assumption that $|e_k|, \ldots, |e_i| \geq \frac{1}{2}$ and either $k = n$ or $|e_{k+1}| < \frac{1}{2}$ we have*

$$\beta_i^2 = \frac{\prod_{j=i}^{k} \frac{j^2}{4}}{\prod_{j=i}^{k} \left(\frac{j^2}{4} - \frac{1}{4}\right)} - \sum_{j=i+1}^{k} \left(e_j^2 \frac{\prod_{l=i}^{j-1} \frac{l^2}{4}}{\prod_{l=i}^{j} \left(\frac{l^2}{4} - \frac{1}{4}\right)} \right) - e_i^2 \frac{1}{\frac{i^2}{4} - \frac{1}{4}}.$$

Proof: We go by reverse induction on i. The case $i = k$ follows by definition. Assume the result holds for $i + 1$. Then

$$\beta_i^2 = \alpha_i^2 \left(\frac{e_i}{\beta_{i+1}} \right) \cdot \beta_{i+1}^2 = \beta_{i+1}^2 \frac{\frac{i^2}{4}}{\frac{i^2}{4} - \frac{1}{4}} - e_i^2 \frac{1}{\frac{i^2}{4} - \frac{1}{4}}.$$

Plugging in β_{i+1}^2 immediately gives the result. □

From $\|\mathbf{e}^{(i)}\|^2 \leq \beta_{i+1}^2 \mu^{(i)2}$ and the Transference Theorems we obtain $e_i^2 \leq \frac{i^2}{4} \beta_{i+1}^2$. So under the condition that $|e_k|, \ldots, |e_{i+1}| > \frac{1}{2}$ we have that

$$e_i^2 \leq \frac{i^2}{4} \left(\frac{\prod_{j=i+1}^{k} \frac{j^2}{4}}{\prod_{j=i+1}^{k} \left(\frac{j^2}{4} - \frac{1}{4}\right)} - \sum_{j=i+2}^{k} \left(e_j^2 \frac{\prod_{l=i+1}^{j-1} \frac{l^2}{4}}{\prod_{l=i+1}^{j} \left(\frac{l^2}{4} - \frac{1}{4}\right)} \right) - e_{i+1}^2 \frac{1}{\frac{(i+1)^2}{4} - \frac{1}{4}} \right).$$

The following corollary follows immediately:

Corollary 3.5.5 *If $|e_k|, \ldots, |e_{i+1}| \geq \frac{1}{2}$ for $1 \leq i < k \leq n$, then*

$$e_i^2 + \sum_{j=i+1}^{k} \left(e_j^2 \frac{\prod_{l=i}^{j-1} \frac{l^2}{4}}{\prod_{l=i+1}^{j} \left(\frac{l^2}{4} - \frac{1}{4}\right)} \right) \leq \frac{\prod_{j=i}^{k} \frac{j^2}{4}}{\prod_{j=i+1}^{k} \left(\frac{j^2}{4} - \frac{1}{4}\right)}. \tag{3.5.11}$$

Recursively, for $i = n, \ldots, 1$, define the sets

$$\begin{aligned} \mathbf{E}_i^{(n)} &:= \left\{ (e_i, \ldots, e_n) \in \mathbb{R}^{n-i+1} : |e_i| \leq \frac{i}{2} \beta_{i+1} \quad \text{and} \quad (e_{i+1}, \ldots, e_n) \in \mathbf{E}_{i+1}^{(n)} \right\} \\ &= \left\{ (e_i, \ldots, e_n) \in \mathbb{R}^{n-i+1} : |e_j| \leq \frac{j}{2} \beta_{j+1} \quad \text{for} \quad j = i, \ldots, n \right\}. \end{aligned}$$

where β_j is defined as in Definition 3.5.1. We will now upper bound the volume of $\mathbf{E}_i^{(n)}$ by first upper bounding the volume of $\mathbf{E}_1^{(n)}$ and then showing that the same bound is basically valid for $\mathbf{E}_i$. Consider the following partition of $\mathbf{E}_1^{(n)}$ into disjunct sets according to the largest index τ such that $e_\tau < 1/2$:

$$\begin{aligned} \mathbf{E}_1^{(n)} &= \underbrace{\left\{ (e_1, \ldots, e_n) \in \mathbb{R}^n : |e_j| \leq \frac{j}{2} \cdot \beta_{j+1} \quad \text{and} \quad |e_j| \geq 1/2, j = 2, \ldots, n \right\}}_{=: \mathbf{T}_1} \\ &\sqcup \bigsqcup_{2 \leq \tau \leq n} \underbrace{\left\{ (e_1, \ldots, e_n) \in \mathbb{R}^n : |e_j| \leq \frac{j}{2} \cdot \beta_{j+1} \quad \text{and} \quad \max_{2 \leq j \leq n} \{ j : |e_j| < 1/2 \} = \tau \right\}}_{=: \mathbf{T}_\tau}. \end{aligned}$$

Hence the volume of $\mathbf{E}_1^{(n)}$ equals the sum of the volumes of $\mathbf{T}_\tau$ on the right hand side. Note that if $(e_1,\dots,e_n) \in \mathbf{T}_\tau$ for $\tau \geq 2$, then the following conditions are satisfied: $(e_{\tau+1},\dots,e_n)$ satisfy Equation (3.5.11) and $(e_1,\dots,e_{\tau-1}) \in \mathbf{E}_1^{(\tau-1)}$. With

$$V_{j,n} := \begin{cases} 1 & \text{if} \quad j = n, \\ \mathrm{vol}_{n-j}\left\{(e_{j+1},\dots,e_n) \in \mathbb{R}^{n-j+1} : \text{ Eq. (3.5.11) holds}\right\} & \text{else,} \end{cases}$$

for $\tau \geq 2$ we have that

$$\mathrm{vol}(\mathbf{T}_\tau) \leq \mathrm{vol}\left(\mathbf{E}_1^{(\tau-1)}\right) V_{\tau,n}.$$

If $(e_1,\dots,e_n) \in \mathbf{T}_\tau$ for $\tau \geq 1$, then $(e_{\tau+1},\dots,e_n)$ satisfy Equation (3.5.11) and $|e_1| \leq 1/2$. Hence $\mathrm{vol}(\mathbf{T}_1) \leq V_{1,n}$. Set $a^{(0)} := 1$ and recursively define

$$a^{(k)} = \sum_{1\leq j\leq k} a^{(j-1)} V_{j,k}. \tag{3.5.12}$$

Then $\mathrm{vol}\big(\mathbf{E}_1^{(n)}\big) \leq a^{(n)}$. Using the well known formula for the volume of an ellipsoid, $V_{j,k}$ can be computed (see Section 3.7.1) to be

$$V_{j,k} = \frac{\pi^{(k-j)/2}}{\Gamma\left(\frac{k-j}{2}+1\right)} \frac{k!}{j!2^{k-j}} \left(\frac{k+1}{j+1}\right)^{1/2} \left(\frac{k}{k+1}\right)^{(k-j)/2},$$

and

$$V_{j,k} \leq \left(\frac{\pi}{4}\right)^{(k-j)/2} \frac{1}{\Gamma\left(\frac{k-j}{2}+1\right)} \frac{k!}{j!} \left(\frac{k+1}{j+1}\right)^{1/2}.$$

Plugging this into Equation (3.5.12), for $k = 1,\dots,n$ we get

$$a^{(k)} \leq \sum_{j=1}^{k} a^{(j-1)} \left(\frac{\pi}{4}\right)^{(k-j)/2} \frac{1}{\Gamma\left(\frac{k-j}{2}+1\right)} \frac{k!}{j!} \left(\frac{k+1}{j+1}\right)^{1/2},$$

which leads to

$$\frac{a^{(k)}}{\sqrt{k+1}(k+1)!} \left(\frac{4}{\pi}\right)^{k/2} \leq \frac{1}{k+1}\sqrt{\frac{4}{\pi}} \sum_{j=1}^{k} \frac{a^{(j-1)}}{\sqrt{j}j!} \left(\frac{4}{\pi}\right)^{(j-1)/2} \frac{1}{\Gamma\left(\frac{k-j}{2}+1\right)}. \tag{3.5.13}$$

We will now derive a nicer recursion, the goal to upper bound $a^{(k)}$ remains the same however. Set $s^{(0)} := 1$ and

$$s^{(k)} := \frac{1}{k+1}\sqrt{\frac{4}{\pi}} \sum_{j=1}^{k} \frac{s^{(j-1)}}{\Gamma\left(\frac{k-j}{2}+1\right)}.$$

Lemma 3.5.6 *Using the notation as above, for all $k \geq 0$ we have that*

$$a^{(k)} \leq \sqrt{k+1}(k+1)! \left(\frac{\pi}{4}\right)^{k/2} s^{(k)}.$$

Proof: The proof goes by induction on k. Clearly $a^{(0)} = 1 = s^{(0)}$. For $k \geq 1$, by Equation (3.5.13) we have

$$\begin{aligned} a^{(k)} &\leq \sqrt{k+1}(k+1)! \left(\frac{\pi}{4}\right)^{k/2} \frac{1}{k+1} \sqrt{\frac{4}{\pi}} \sum_{j=1}^{k} \frac{a^{(j-1)}}{\sqrt{j} j!} \left(\frac{4}{\pi}\right)^{(j-1)/2} \frac{1}{\Gamma\left(\frac{k-j}{2}+1\right)} \\ &\leq \sqrt{k+1}(k+1)! \left(\frac{\pi}{4}\right)^{k/2} \frac{1}{k+1} \sqrt{\frac{4}{\pi}} \sum_{j=1}^{k} s^{(j-1)} \frac{1}{\Gamma\left(\frac{k-j}{2}+1\right)} \\ &= \sqrt{k+1}(k+1)! \left(\frac{\pi}{4}\right)^{k/2} s^{(k)}. \end{aligned}$$

□

So any upper bound on $s^{(n)}$ directly gives an upper bound on $a^{(n)}$. Unfortunately despite the seemingly nice structure of the recursively defined sequence $s^{(n)}$, deriving a provable explicit upper bound on $s^{(n)}$ seems a nontrivial task. Numerical computation of $s^{(n)}$ showed (compare Figure 1) that for $10 < n \leq 2000$ we have

$$\underbrace{\frac{\log\left(s^{(n)}\sqrt{n+1}(n+1)!\,(\pi/4)^{n/2}\right)}{n \log n}}_{=:c_n} < 0.75.$$

As

$$\frac{\log a^{(n)}}{n \log n} \leq \frac{\log\left(s^{(n)}\sqrt{n+1}(n+1)!\,(\pi/4)^{n/2}\right)}{n \log n} = c_n,$$

it follows that

$$\mathrm{vol}\left(\mathbf{E}_1^{(n)}\right) = a^{(n)} \leq n^{c_n n}$$

and hence for $10 < n \leq 2000$

$$\mathrm{vol}\left(\mathbf{E}_1^{(n)}\right) < n^{0.75n}.$$

It remains to show that this bound basically also holds for $\mathrm{vol}\left(\mathbf{E}_i^{(n)}\right)$, where $1 \leq i \leq n$, i.e. that

$$\mathrm{vol}\left(\mathbf{E}_i^{(n)}\right) \leq a^{(n)}.$$

In fact this is true:

Lemma 3.5.7 *Using the notation from above we have that*

$$\mathrm{vol}\left(\mathbf{E}_i^{(n)}\right) \leq 2a^{(n)}.$$

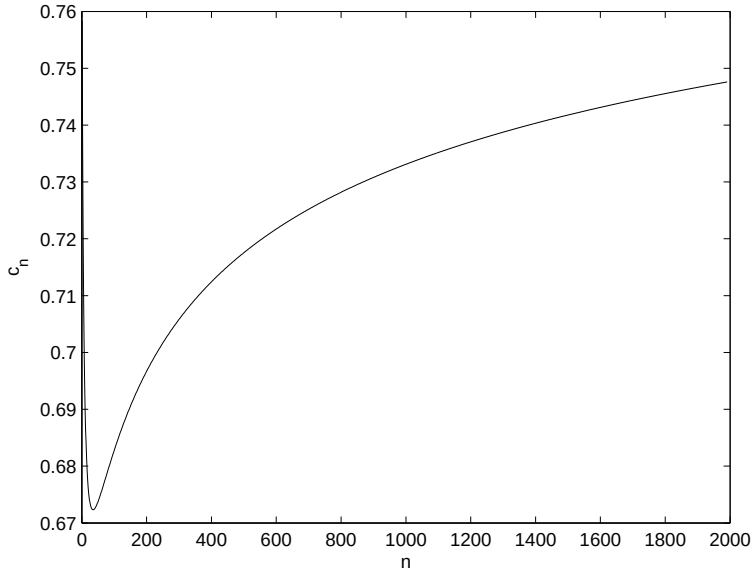


Figure 1: The behaviour of c_n for $10 \leq n \leq 2000$.

Proof: For $i \geq 1$ set $a_i^{(i-1)} := i$ and recursively define

$$a_i^{(k)} = \sum_{j=i}^{k} a_i^{(j-1)} V_{j,k}.$$

Then $\mathrm{vol}\left(\mathbf{E}_i^{(n)}\right) \leq a_i^{(n)}$. Note that for $i = 1$, this definition coincides with the one of $a^{(k)} = a_1^{(k)}$. For $i \geq 3$, we have

$$a^{(i-1)} = \sum_{j=1}^{i-1} a^{(j-1)} V_{j,i-1} \geq a^{(i-3)} V_{i-2,i-1} + a^{(i-2)} V_{i-1,i-1} \geq i - 1 + 1 = i,$$

and hence $a^{(i-1)} \geq i = a_i^{(i-1)}$. From the recursive definition it follows that $a^{(n)} \geq a_i^{(n)}$ for $3 \leq i \leq n$. Let us consider the case $i = 2$. We have $a^{(i-1)} = a^{(1)} = 1$ and $a_i^{(i-1)} = a_2^{(1)} = 2$. From the recursive definitions we can conclude that $2a^{(n)} \geq 2a_2^{(n)}$ for $n \geq 2$. □

Note that

$$a^{(n)} = \sum_{1 \leq j \leq n} a^{(j-1)} V_{j,n} \geq a^{(n-2)} V_{n-1,n} = n \cdot a^{(n-2)}.$$

As a consequence we have the following lower bound on $a^{(n)}$:

$$\mathrm{vol}\left(\mathbf{E}_1^{(n)}\right) = a^{(n)} \geq \sqrt{n!} > n^{0.5(n - n/\ln n)}.$$

3.6 Conclusion

We have seen that given a dual HKZ-basis, we can solve the closest vector problem using a refined version of the approach by Blömer [Blö00] by enumerating an expected number of $n^{c_n n}$ lattice points, with $c_n < 0.75$ for $10 < n \leq 2000$. Kannan's algorithm runs faster, as refined analysis thereof implies [HS10]. Using Kannan's algorithm, which as input takes an HKZ-basis, it is enough to enumerate $n^{n/2+o(n)}$ lattice points. Table 1 gives an overview on the complexities.

Approach	original	refined
Kannan	$n^{n+o(n)}$	$2^{\mathcal{O}(n)} n^{n/2}$
Blömer	$n!$	$n^{c_n n}$

Table 1: Overview on the expected number of points to enumerate.

3.7 Appendix

3.7.1 Computation of $V_{\tau,k}$ in Section 3.5

$$
\begin{aligned}
V_{\tau,k} &= \frac{\pi^{(k-\tau)/2}}{\Gamma\left(\frac{k-\tau}{2}+1\right)} \left(\frac{\prod_{j=\tau+1}^{k} \frac{j}{2}}{\prod_{j=\tau+2}^{k} \left(\frac{j^2}{4}-\frac{1}{4}\right)^{1/2}} \right)^{k-\tau+1} \prod_{j=\tau+2}^{k} \frac{\prod_{i=\tau+2}^{j} \left(\frac{i^2}{4}-\frac{1}{4}\right)^{1/2}}{\prod_{i=\tau+1}^{j-1} \frac{i}{2}} \\
&= \frac{\pi^{(k-\tau)/2}}{\Gamma\left(\frac{k-\tau}{2}+1\right)} \left(\frac{\tau+1}{2}\right)^{k-\tau} \left(\prod_{j=\tau+2}^{k} \frac{\frac{j}{2}}{\sqrt{\frac{j^2}{4}-\frac{1}{4}}} \right)^{k-\tau} \\
&\quad \cdot \frac{k!}{(\tau+1)^{k-\tau-1}(\tau+1)!} \prod_{j=\tau+2}^{k} \prod_{i=\tau+2}^{j} \frac{\left(\frac{i^2}{4}-\frac{1}{4}\right)^{\frac{1}{2}}}{\frac{i}{2}} \\
&= \frac{\pi^{(k-\tau)/2}}{\Gamma\left(\frac{k-\tau}{2}+1\right)} \frac{k!}{\tau! 2^{k-\tau}} \left(\frac{(\tau+2)k}{(\tau+1)(k+1)}\right)^{(k-\tau)/2} \left(\frac{\tau+1}{\tau+2}\right)^{(k-\tau-1)/2} \left(\frac{k+1}{\tau+2}\right)^{1/2} \\
&= \frac{\pi^{(k-\tau)/2}}{\Gamma\left(\frac{k-\tau}{2}+1\right)} \frac{k!}{\tau! 2^{k-\tau}} \left(\frac{k+1}{\tau+1}\right)^{1/2} \left(\frac{k}{k+1}\right)^{(k-\tau)/2} \\
&\leq \frac{\pi^{(k-\tau)/2}}{\Gamma\left(\frac{k-\tau}{2}+1\right)} \frac{k!}{\tau! 2^{k-\tau}} \left(\frac{k+1}{\tau+1}\right)^{1/2} \\
&= \left(\frac{\pi}{4}\right)^{(k-\tau)/2} \frac{1}{\Gamma\left(\frac{k-\tau}{2}+1\right)} \frac{k!}{\tau!} \left(\frac{k+1}{\tau+1}\right)^{1/2}.
\end{aligned}
$$

Chapter 4

Improvements in the AKS sieve

For simplicity we consider only full rank lattices $\mathcal{L} \subset \mathbb{R}^n$ in this chapter. Algorithms to solve the shortest vector problem exactly can roughly be divided into two main branches, enumeration and sieving. Enumeration searches all lattice points in some appropriate subset of the Euclidean space containing the shortest vectors of the lattice. The time complexity is often estimated by the Gaussian heuristic and the volume of the subset searched (cf. Section 3.1). The space complexity is basically linear in the dimension n. Enumeration is still the most used algorithm when it comes to finding a shortest lattice vector in low dimensions and practical enumeration algorithms (see e.g. [SE94]) run in time $2^{\mathcal{O}(n^2)}$, the best asymptotic bound being $2^{\mathcal{O}(n \log n)}$ [Kan83].

In 2001, Ajtai, Kumar and Sivakumar came up with an asymptotically faster algorithm, the AKS sieve algorithm [AKS01]. The AKS sieve algorithm provably runs in probabilistic time $2^{\mathcal{O}(n)}$. A main drawback of the algorithm is the fact that also the space requirement is $2^{\mathcal{O}(n)}$ and so far the AKS algorithm is still outperformed in practice by state of the art enumeration algorithms. In [NV08], careful analysis of the original AKS sieve revealed the hidden constants in the big-$\mathcal{O}$ notation showing that with an appropriate choice of parameters the AKS sieve algorithm runs in time $2^{5.9n+\mathcal{O}(\log n)}$ and space $2^{2.95n+\mathcal{O}(\log n)}$. A main ingredient in their analysis is a bound on the maximum number of points of mutual distance $\lambda > 0$ inside a ball of radius $R > 0$. They use the fact that this number can be bounded by 2^{cn}, with $c = \log_2(1 + 2R/\lambda)$. We will show that this bound can in fact be reduced to $2^{c'n+\mathcal{O}(\log n)}$, with $c' = \log_2(\sqrt{2}R/\lambda)$, by combining a result on sphere packings by [MG02] and a result on the minimum number of balls needed to cover a larger ball [Rog63]. Redoing the complexity analysis for the AKS algorithm we will show that the time and space complexity can be bounded by $2^{4.2n+\mathcal{O}(\log n)}$, $2^{2.1n+\mathcal{O}(\log n)}$ respectively.

The decisions made in the AKS algorithm are not based on the given set of lattice vectors directly, but on perturbations thereof. This special perturbation method is used to prove that with high probability a shortest vector is returned. In [NV08] it is argued that this perturbation might be omitted in practice. They propose a new sieving algorithm that —under a given heuristic assumption— runs in time $\text{poly}(n)(4/3 + \epsilon)^n$ with space requirement $\text{poly}(n)(4/3 + \epsilon)^{n/2}$. We review this algorithm and propose

a generalization thereof reducing the theoretical gap between enumeration algorithms and sieving. While the original sieving algorithm tries to reduce the lengths of a large set of lattice vectors by finding shorter vectors in sublattices of rank 2, we propose an algorithm that reduces the lengths of the set of lattice vectors by running an SVP-solver (e.g. enumeration) on sublattices of arbitrary higher ranks (Section 4.2.1). A thorough analysis of this approach is still to be done, kick-off considerations are given though. First experiments show that this approach could make sieving practical to higher dimensions than to date.

The chapter is organized as follows: In Section 4.1 we describe the AKS sieving algorithm including the complexity analysis by [NV08] in Section 4.1.1. In Section 4.1.2 we derive a new bound on the number of lattice points inside a ball of given radius and in Section 4.1.3 we show how the parameters of the AKS algorithm can be adapted resulting in better complexity upper bounds. In Section 4.2 we review the heuristic sieve algorithm proposed by [NV08] and present a generalization thereof in Section 4.2.1.

4.1 AKS sieving

The principle of classical AKS sieving algorithm [AKS01] to solve the shortest vector problem is as follows:

1. *Sample* a large set $\mathbf{S}$ of lattice vectors inside a ball of given radius around the origin.

2. Iteratively apply a *sieving step* on $\mathbf{S}$ to obtain a new large set of lattice vectors inside a ball of smaller radius.

3. Compute the *pairwise differences* between the vectors in the resulting set to get a shortest vector.

The algorithm is a probabilistic algorithm. In order to be able to prove that the algorithm with high probability returns the desired result, a perturbation method is applied. This means that the decisions made in the algorithm do not depend on the lattice vectors $\mathbf{v} \in \mathcal{L}$ sampled, but instead on some random perturbations $\mathbf{y} \in \mathbb{R}^n$ thereof.

In the following the different parts of the sieving algorithm are described following the work in [NV08]. We will not repeat the algorithm in full detail. The goal is to provide enough details such that the essence of the complexity analysis can be understood.

The algorithm depends on three input parameters $\gamma, \xi, c_0 \in \mathbb{R}$. The shrinking parameter γ is chosen to be in $(0, 1)$ and determines by which factor the radius of the ball is reduced in every sieving step. The second parameter ξ is of the form $\xi' \lambda_1(\mathcal{L})$ with $\xi' > 1/2$ and bounds the perturbation applied to the lattice vectors. Note that $\lambda_1(\mathcal{L})$ is not known in advance and is hard to find in general. However for $\alpha > 1$, using the LLL reduction algorithm we can find in polynomial time a vector $\mathbf{v} \in \mathcal{L}$ such that $\lambda_1(\mathcal{L}) \leq \mathbf{v} \leq \alpha^{\mathrm{poly}(n)} \lambda_1(\mathcal{L})$. Making a polynomial number of guesses we find a good approximation of λ such that $\lambda_1(\mathcal{L}) \leq \lambda \leq \alpha \lambda_1(\mathcal{L})$. Finally, c_0 determines the

number $N_0 = 2^{c_0 n}$ of vectors to be sampled. The space requirement of the algorithm will be dominated by N_0 and the time complexity is dominated by taking the pairwise differences of the vectors resulting in roughly N_0^2 polynomial time operations.

Initial sampling The sieving algorithm first computes a LLL reduced basis for which it can be shown that $\max_{i=1,\ldots,n} \|\mathbf{b}_i\| \leq \beta^n \lambda_1(\mathcal{L})$ for some $\beta > 1$, unless a shortest vector is contained in a proper sublattice of the form $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$, $k < n$ (cf. [NV08, Lemma 3.3]). Let $R_0 = n \max_i \|\mathbf{b}_i\|$. The initial sampling procedure generates a set $\mathbf{S}_0$ of $N_0 = 2^{c_0 n}$ pairs $(\mathbf{v}, \mathbf{y}) \in \mathcal{L} \times \mathcal{B}_n(R_0)$, such that $\mathbf{v} - \mathbf{y}$ is uniformly distributed in $\mathcal{B}_n(\xi)$. This is done by applying Babai's nearest plane algorithm [Bab86] to random vectors from $\mathcal{B}_n(\xi)$.

Sieving step Taking as input a set $\mathbf{S}_i$, $i \geq 0$, of N_i pairs $(\mathbf{v}, \mathbf{y}) \in \mathcal{L} \times \mathcal{B}_n(R_i)$, and setting $R_{i+1} = \gamma R_i + \xi$, it outputs a set $\mathbf{S}_{i+1}$ of N_i pairs $(\mathbf{v}, \mathbf{y}) \in \mathcal{L} \times \mathcal{B}_n(R_{i+1})$ using the procedure as described in Algorithm 8. It is easy to see that $R_i = \gamma^i R_0 + \xi \frac{1-\gamma^i}{1-\gamma}$, that the cardinalities of the sets $\mathbf{S}$, $\mathbf{S}'$ and $\mathbf{C}$ from Algorithm 8 satisfy

$$|\mathbf{S}| = |\mathbf{S}'| + |\mathbf{C}|,$$

and that $|\mathbf{C}|$ is bounded by the maximum number of elements at mutual distance at least γR inside the annulus $\{\mathbf{x} \in \mathbb{R}^n : \gamma R \leq \|\mathbf{x}\| \leq R\}$. This will be crucial in the complexity analysis.

Algorithm 8: Sieving step

Input: A set $\mathbf{S}$ of pairs $(\mathbf{v}, \mathbf{y}) \in \mathcal{L} \times \mathcal{B}_n(R)$ where $\mathbf{v} - \mathbf{y}$ is uniformly at random in $\mathcal{B}_n(\xi)$ and $\gamma \in (0, 1)$.
Output: A set $\mathbf{S}'$ of pairs $(\mathbf{v}, \mathbf{y}) \in \mathcal{L} \times \mathcal{B}_n(\gamma R + \xi)$.

```
S' ← ∅
C ← ∅
for (v, y) ∈ S do
    if ∃(v', y') ∈ C s.t. ‖y − y'‖ ≤ γR then
        S ← S ∪ (v − v', y − v')
    else
        C ← C ∪ (v, y)
    end
end
return S'
```

Pairwise differences Let $\mathbf{S}_l \subset \mathcal{L} \times \mathcal{B}_n(R_l)$ be the set of pairs obtained by iteratively applying the sieving step l times to an initially sampled set $\mathbf{S}_0 \subset \mathcal{L} \times \mathcal{B}_n(R_0)$. In this case $R_l = \gamma^l R_0 + \xi \frac{1-\gamma^l}{1-\gamma}$. Note that for $(\mathbf{v}, \mathbf{y}) \in \mathbf{S}_l$, by the triangle inequality the length

of the lattice vector $\mathbf{v}$ is bounded:

$$\|\mathbf{v}\| \leq \underbrace{\gamma^l R_0 + \xi \frac{1-\gamma^l}{1-\gamma} + \xi}_{=R_l+\xi}.$$

In the limit $l \to \infty$, R_l tends to $\xi\frac{1}{1-\gamma}$ and for e.g. $k = \lceil \log_\gamma \frac{0.01\xi}{R_0(1-\gamma)} \rceil$ we get close to the limit, i.e. $R_k \leq \xi\frac{1.01}{1-\gamma}$. If $|\mathbf{S}_k|$ is large enough we expect to find a shortest vector of $\mathcal{L}$ by taking the pairwise difference of the elements in $\mathbf{S}_k$ (see Lemma 4.1.1).

4.1.1 Complexity analysis for AKS

We will now sketch the complexity analysis done by Nguyen et al [NV08]. In their analysis they use the following notation.

1. $2^{c_R n}$: denotes an upper bound on the number of lattice points inside a ball of radius R. By [NV08, Lemma 3.2],

$$c_R = \log_2\left(1 + \frac{2R}{\lambda_1(\mathcal{L})}\right). \tag{4.1.1}$$

2. $2^{c_S n}$: denotes an upper bound on the number of points with mutual distance at least γR inside an n-dimensional annulus $\{\mathbf{x} \in \mathbb{R}^n : \gamma R \leq \|\mathbf{x}\| \leq R\}$. By [NV08, Lemma 3.5],

$$c_S = \begin{cases} \log_2(2/\gamma) & \text{if } \gamma \geq \frac{1}{2}\left(\sqrt{5}-1\right), \\ \frac{1}{2}\log_2(2 + \frac{2}{2\gamma-1}) & \text{if } \gamma < \frac{1}{2}\left(\sqrt{5}-1\right). \end{cases} \tag{4.1.2}$$

3. $2^{-c_U n}$: let $\xi > \lambda_1(\mathcal{L})/2$, and $\mathbf{y} \in \mathcal{L}$ such that $\|\mathbf{y}\| = \lambda_1(\mathcal{L})$. Then $2^{-c_U n}$ is a lower bound on the probability that for an element $\mathbf{x}$ chosen uniformly at random from $\mathcal{B}_n(\xi)$ also $\mathbf{x}+\mathbf{y}$ is in $\mathcal{B}_n(\xi)$. By [NV08, Lemma 3.4],

$$c_U = \frac{1}{2}\log_2 \frac{\xi^2}{\xi^2 - \lambda_1^2/4} = \frac{1}{2}\log_2 \frac{\xi'^2}{\xi'^2 - 1/4}. \tag{4.1.3}$$

Lemma 4.1.1 ([NV08] Lemma 3.6) *Let $\mathcal{L} \in \mathbb{R}^n$ be an n-dimensional lattice, (γ, ξ, c_0) a choice of parameters for the sieving algorithm as above, $k = \lceil \log_\gamma \frac{0.01\xi}{R_0(1-\gamma)} \rceil$ and $R_\infty = \xi\left(1 + \frac{1.01}{1-\gamma}\right)$. Let c_{R_∞}, c_U and c_S be as in* (4.1.1), (4.1.3), (4.1.2) *and*

$$N_\infty = 2^{(c_0 - c_U)n - 1} - k2^{c_S n}.$$

Then if $N_\infty \geq 2^{c_{R_\infty} n + 3}$, the sieving algorithm outputs a shortest vector of $\mathcal{L}$ with probability at least $1/2$.

Lemma 4.1.1 imposes the following constraints on (γ, ξ, c_0) (compare [NV08, page 12]):

$$c_0 - c_U > c_S, \tag{4.1.4}$$

$$c_0 - c_U > c_{R_\infty}. \tag{4.1.5}$$

Optimizing the parameters with respect to the running time, Nguyen et al propose to use the following parameters such that a shortest vector of the lattice is found by the algorithm with high probability:

$$(\gamma, \xi, c_0) = \big(0.518, 0.7\lambda_1(\mathcal{L}), 2.95\big).$$

The running time of the algorithm is dominated by taking the pairwise differences resulting in $\mathrm{poly}(n)2^{2c_0 n}$ polynomial time operations. The space complexity is dominated by the pairs initially sampled, i.e. by $\mathrm{poly}(n) \cdot 2^{c_0 n}$ bits. See Table 1 for an overview on the variables.

(γ, ξ, c_0)	c_{R_∞}	c_S	c_U	time	space
$(0.518, 0.7\lambda_1, 2.95)$	2.42	0.79	0.51	$2^{5.9n+\mathcal{O}(\log n)}$	$2^{2.95n+\mathcal{O}(\log n)}$

Table 1: Crucial values for the given choices of parameters.

We will improve these complexity bounds in Section 4.1.3. The improvement is based on new upper bounds on the number of lattice points inside a ball of given radius derived in the next section.

4.1.2 New bounds on the number of lattice points inside a ball

In this section a new bound on the number of points with given minimal mutual distance inside a ball of radius $R > 0$ is given. The bound follows from a result by Rogers [Rog63], bounding the minimal number of balls of radius 1 required to cover a ball of radius R and a result by Micciancio and Goldwasser [MG02] bounding the number of points with mutual distance at least 2 inside a ball of radius $\sqrt{2}$.

Let $N_{R,n}$ denote the minimal number of n-balls of radius 1 needed to cover an n-ball of radius R. I.e. there exists a set $\mathbf{X} \subset \mathbb{R}^n$ with $|\mathbf{X}| = N_{R,n}$ such that

$$\mathcal{B}_n(R) \subseteq \bigcup_{x \in \mathbf{X}} \mathcal{B}_n(x, 1).$$

Rogers [Rog63, Proof of Thm. 3] (see also [VG05] for an improvement) derived the following bounds on $N_{R,n}$.

Theorem 4.1.2 *Let $R > 1$ and $n \geq 9$. Using the notation from above we have that*

$$N_{R,n} \leq \begin{cases} e(n \ln n + n \ln \ln n + 5n)R^n & \text{if} \quad R \geq n, \\ n(n \ln n + n \ln \ln n + 5n)R^n & \text{if} \quad \frac{n}{\ln n} \leq R < n, \\ \frac{4en\sqrt{n}}{\ln n - 2}\big(2n \ln n + n \ln \ln n + \frac{1}{2}\ln 144n\big)R^n & \text{if} \quad R < \frac{n}{\ln n}. \end{cases}$$

We further have the following theorem on sphere covering:

Theorem 4.1.3 ([MG02], Theorem 5.2) *The maximum number of points in a sphere of radius $\frac{1}{\sqrt{2}}$ in $\mathbb{R}^n$ with mutual distance at least 1 to each other is $2n$.*

These two results can be used to bound the number of points with mutual distance at least λ inside a sphere of radius R.

Lemma 4.1.4 *Let R, λ be positive reals and $\mathbf{S} \subset \mathcal{B}_n(R)$ a set of points with mutual distance at least λ. Then with $R' = \frac{\sqrt{2}}{\lambda}R$, we have that*

$$|\mathbf{S}| \leq 2nN_{R',n}.$$

If $n \geq 9$,

$$|\mathbf{S}| \leq 2nN_{R',n} = \text{poly}(n)\left(\frac{\sqrt{2}}{\lambda}R\right)^n.$$

Proof: Note that the minimum number of balls of radius $\frac{\lambda}{\sqrt{2}}$ required to cover a ball of radius R equals the minimum number of balls of radius 1 required to cover a ball of radius $\frac{\sqrt{2}}{\lambda}R$. Hence $\mathcal{B}_n(R)$ can be covered by $N_{R',n}$ balls of radius $\frac{\lambda}{\sqrt{2}}$. By Theorem 4.1.3 each of these balls contains at most $2n$ points with mutual distance at least λ. The second part follows from Theorem 4.1.2. □

Setting $\lambda = \gamma R$ with $0 < \gamma < 1$ we immediately get:

Corollary 4.1.5 *Let $R \in \mathbb{R}$ be positive and $0 < \gamma < 1$. Let $\mathbf{C} \subset \mathcal{B}_n(R)$ a set of points with mutual distance at least γR. Then with $R' = \frac{\sqrt{2}}{\gamma}$, we have that*

$$|\mathbf{C}| \leq 2nN_{R',n}.$$

If $n \geq 9$,

$$|\mathbf{C}| \leq 2nN_{R',n} = \text{poly}(n)\left(\frac{\sqrt{2}}{\gamma}\right)^n.$$

In the next section we see how these bounds influence the complexity upper bounds for the sieving algorithm.

4.1.3 New complexity upper bounds for AKS

Using the results from Section 4.1.2 we will show how the complexity upper bounds on the AKS algorithm as given in Table 1 can be improved in the case where $n \geq 9$.

1. $2^{\hat{c}_R n}$: denotes an upper bound on the number of lattice points inside a ball of radius R. By Lemma 4.1.4, with $R' = \frac{\sqrt{2}}{\lambda_1(\mathcal{L})}R$ we can set

$$\hat{c}_R = \frac{1}{n}\log_2\left(2nN_{R',n}\right) = \frac{\mathcal{O}\left(\log n\right)}{n} + \log\frac{\sqrt{2}R}{\lambda_1(\mathcal{L})}.$$

2. $2^{\hat{c}_S n}$: denotes an upper bound on the number of points inside an n-dimensional annulus $\{\mathbf{x} \in \mathbb{R}^n : \gamma R \leq \|\mathbf{x}\| \leq R\}$ with mutual distance at least γR. By Corollary 4.1.5, with $R' = \frac{\sqrt{2}}{\gamma}$ we can set

$$\hat{c}_S = \frac{1}{n} \log_2 \left(2nN_{R',n}\right) = \frac{\mathcal{O}\left(\log n\right)}{n} + \log \frac{\sqrt{2}}{\gamma}.$$

As in Section 4.1.1 we can state the following constraints on a parameter triple (γ, ξ, c_0) (compare (4.1.4) and (4.1.5)):

$$c_0 - c_U > \hat{c}_S, \tag{4.1.6}$$
$$c_0 - c_U > \hat{c}_{R_\infty}. \tag{4.1.7}$$

The goal is to find a optimal valid parameter triple (γ, ξ, c_0) such that these constraints are satisfied. Recall that $R_\infty = \xi \cdot \left(1 + \frac{1.01}{1-\gamma}\right) = \lambda_1 \cdot \xi' \cdot \left(1 + \frac{1.01}{1-\gamma}\right)$ and hence we get

$$\begin{aligned}
\hat{c}_{R_\infty} &= \frac{1}{n} \log_2 \left(2nN_{R_1,n}\right) \quad \text{with} \quad R_1 = \sqrt{2} \cdot \xi' \cdot \left(1 + \frac{1.01}{1-\gamma}\right), \\
\hat{c}_S &= \frac{1}{n} \log_2 \left(2nN_{R_2,n}\right) \quad \text{with} \quad R_2 = \frac{\sqrt{2}}{\gamma}, \\
c_U &= \frac{1}{2} \log_2 \frac{\xi'^2}{\xi'^2 - 1/4}.
\end{aligned}$$

Note that for a fixed ξ', $\hat{c}_S$ is decreasing in γ and $\hat{c}_{R_\infty}$ is increasing in γ. As c_U is independent of γ, we choose γ such that $\hat{c}_S = \hat{c}_{R_\infty}$ or equivalently that $R_1 = R_2$. Solving the corresponding quadratic equation and keeping in mind that $\gamma \in (0, 1)$ we get

$$\gamma = \frac{2.01\xi' + 1 - \sqrt{(2.01\xi' + 1)^2 - 4\xi'}}{2\xi'}.$$

For this choice of γ it remains to minimize $c_U + \hat{c}_S$, or equivalently $c_U + \hat{c}_{R_\infty}$, with respect to ξ'. Note that $\hat{c}_S = \log_2 R_2 + \frac{\mathcal{O}(\log_2 n)}{n}$, where the second summand is independent of ξ'. We minimize

$$\log_2 R_2 + c_U = \log_2 \frac{\sqrt{2} \cdot 2\xi'}{2.01\xi' + 1 - \sqrt{(2.01\xi' + 1)^2 - 4\xi'}} + \frac{1}{2} \log_2 \frac{\xi'^2}{\xi'^2 - 1/4}.$$

Numerical computations show that the minimum is reached for $\xi' \approx 0.794$. The corresponding value for the shrinking factor is $\gamma \approx 0.446$. For these values of ξ' and γ we get:

$$\begin{aligned}
\hat{c}_S = \hat{c}_{R_\infty} &= 1.66 + \frac{\mathcal{O}\left(\log n\right)}{n}, \\
c_U &= 0.36.
\end{aligned}$$

Setting $c_0 = 2.1 + \frac{\mathcal{O}(\log n)}{n}$, Equations (4.1.6) and (4.1.7) are satisfied and the resulting time complexity equals

$$\text{poly}(n)2^{2c_0 n} = 2^{4.2n+\mathcal{O}(\log n)},$$

and the space complexity equals

$$\text{poly}(n)2^{c_0 n} = 2^{2.1n+\mathcal{O}(\log n)}.$$

4.2 Heuristic sieve algorithm

In [NV08], a heuristic variant of the AKS sieve algorithm is presented and analyzed. They point out that the perturbation vectors $\mathbf{y}$, while essential in the theoretical proof of the algorithm, might be ignored in practice. Further it is argued that the bound on the number of elements in $\mathbf{C}$ discarded in each sieving step is a worst case bound and that by the birthday paradoxon this number should be substantially smaller in practice. They propose that on a set $\mathbf{S}_0 \subset \mathcal{L} \cap \mathcal{B}_n(R_0)$, with $R_0 = 2^{\mathcal{O}(n)}\lambda_1(\mathcal{L})$, a sieving step as described in Algorithm 9 is iteratively applied until a short vector is found.

Algorithm 9: Pair sieving step

Input: A set $\mathbf{S} \subset \mathcal{L} \cap \mathcal{B}_n(R)$ and $\gamma \in (0, 1)$.
Output: A set $\mathbf{S}' \subset \mathcal{L} \cap \mathcal{B}_n(\gamma R)$.

$\mathbf{S}' \leftarrow \emptyset, \mathbf{C} \leftarrow \emptyset$
for $\mathbf{v} \in \mathbf{S}$ **do**
  if $\|\mathbf{v}\| \leq \gamma R$ **then**
    $\mathbf{S}' \leftarrow \mathbf{S}' \cup \{\mathbf{v}\}$
  else if $\exists \mathbf{c} \in \mathbf{C}$ *s.t.* $\|\mathbf{v} - \mathbf{c}\| \leq \gamma R$ **then**
    $\mathbf{S}' \leftarrow \mathbf{S}' \cup \{\mathbf{v} - \mathbf{c}\}$
  else
    $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mathbf{v}\}$
  end
end
return $\mathbf{S}'$

The analysis of the sieving step as proposed in Algorithm 9 relies on the following assumption

- The elements in $\mathbf{S} \cap \mathcal{C}_n(\gamma, R)$ are uniformly distributed in $\mathcal{C}_n(\gamma, R) = \{x \in \mathbb{R}^n : \gamma R \leq \|x\| \leq R\}$.

Under this assumption on the input set $\mathbf{S}$ the following lemma allows to bound the number of elements discarded in every sieving step.

Lemma 4.2.1 ([NV08] Lemma 4.1) *Let $n \in \mathbb{N}$, $2/3 < \gamma < 1$ and $\mathbf{S}$ as set of points chosen independently uniformly at random from $\mathcal{C}_n(\gamma, R)$. Further let*

$$N_{\mathbf{C}} = \left(\frac{1}{\gamma\sqrt{1-\gamma^2/4}}\right)^n \left\lceil 3\sqrt{2\pi}(n+1)^{3/2} \right\rceil,$$

and $N_{\mathbf{S}} = |\mathbf{S}|$. Then if $N_{\mathbf{C}} < N_{\mathbf{S}} < 2^n$, for any subset $\mathbf{C} \subset \mathbf{S}$ of cardinality at least $N_{\mathbf{C}}$ with elements chosen independently at random, with overwhelming probability, for all $\mathbf{v} \in \mathbf{S}$ there exists a $\mathbf{c} \in \mathbf{C}$ such that $\|\mathbf{v} - \mathbf{c}\| \leq \gamma R$.

Under the given assumption we have that the output set $\mathbf{S}'$ of Algorithm 9 with overwhelming probability has cardinality

$$|\mathbf{S}'| = |\mathbf{S}| - |\mathbf{C}| \geq |\mathbf{S}| - N_{\mathbf{C}},$$

where $N_{\mathbf{C}}$ is as in Lemma 4.2.1. Starting with a sufficiently large set $\mathbf{S}_0 \subset \mathcal{L} \cap \mathcal{B}_n(R)$ of lattice vectors of bounded length $R = 2^{\mathcal{O}(n)}\lambda_1(\mathcal{L})$, after a polynomial number of sieving steps with factor $\gamma < 1$, we can expect to find a shortest vector. Lemma 4.2.1 implies that it is sufficient to choose the cardinality of $\mathbf{S}_0$ to be $\mathrm{poly}(n)N_{\mathbf{C}}$. The running time will be quadratic in the size of the initially sampled set: $\mathcal{O}\left(|\mathbf{S}_0|^2\right)$. By choosing γ close to 1, the space requirement is hence expected to be of order $(4/3+\epsilon)^n$ and the time complexity to be of order $(4/3+\epsilon)^{n/2}$ (cf. [NV08, page 15]).

In the next section we propose an extension of the heuristic sieving algorithm as described above. Instead of taking pairwise differences between the vectors in $\mathbf{S}$ to get potentially shorter vectors, we allow to take i-tuples of vectors and run a SVP-solver on them in order to find the shortest vector in the lattice generated by the given i-tuple. We will prove a result corresponding to Lemma 4.2.1 for this case and discuss the implications on the running time and space requirement.

4.2.1 Extensions of the heuristic sieve

We start by giving the pseudocode of the extended sieve algorithm (see Algorithm 10). Note that when setting the input parameter $\beta = 2$, Algorithm 10 differs from Algorithm 9 only in more sophisticated way to generate new vectors, namely instead of taking the pairwise differences of two lattice vectors, an SVP-solver (Shortest$(\cdot)$) is applied to pairs of basis vectors.

Note that the set $\mathbf{C}$ in the algorithm has the following property:

$$\forall 1 \leq j \leq \beta : \forall \mathbf{c}_1, \ldots, \mathbf{c}_j \in \mathbf{C} : \lambda_1\big(\mathcal{L}(\mathbf{c}_1, \ldots, \mathbf{c}_j)\big) > \gamma R.$$

Clearly $|\mathbf{S}'| = |\mathbf{S}| - |\mathbf{C}|$, and using the same line of thought as in the last section, if $|\mathbf{S}'|$ is not much smaller than the input set $|\mathbf{S}|$, then iteratively applying this sieving step to an initial set $\mathbf{S}_0 \subset \mathcal{L} \cap \mathcal{B}_n(R)$ with $R = 2^{\mathcal{O}(n)}\lambda_1(\mathcal{L})$, after a polynomial number of sieving steps with factor $\gamma < 1$, we can expect to find a shortest vector. It is clear that for $\beta > 2$, given the same input set $\mathbf{S}$ the cardinality of the set $\mathbf{C}$ in Algorithm 10 is smaller

Algorithm 10: Tuple sieving step

Input: A set $\mathbf{S} \subset \mathcal{L} \cap \mathcal{B}_n(R)$ and $\gamma \in (0,1)$, $i \geq 0$.
Output: A set $\mathbf{S}' \subset \mathcal{L} \cap \mathcal{B}_n(\gamma R)$.

$\mathbf{S}' \leftarrow \emptyset, \mathbf{C} \leftarrow \emptyset$
for $\mathbf{v} \in \mathbf{S}$ **do**
  if $\|\mathbf{v}\| \leq \gamma R$ **then**
    $\mathbf{S}' \leftarrow \mathbf{S}' \cup \{\mathbf{v}\}$
  if $\exists (\mathbf{c}_1, \ldots, \mathbf{c}_{i-1}) \in \mathbf{C}^{i-1}$ *s.t.* $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{i-1})\big) \leq \gamma R$ **then**
    $\mathbf{S}' \leftarrow \mathbf{S}' \cup \mathrm{Shortest}\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{i-1})\big)$
  else
    $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mathbf{v}\}$
  end
end
return $\mathbf{S}'$

than the cardinality of the corresponding set in Algorithm 9. On the other hand, the running time of the algorithm will be exponential in β. The ratio of the time-memory tradeoff is not immediately clear.

The question is to what extend the choice of the parameter β influences the size of $\mathbf{C}$. In the following we give first steps towards giving the answer using similar techniques as in [NV08]. A complete analysis is still work to be done. We need the following assumptions, and their correctness is not clear at all.

- If the elements of the input set $\mathbf{S}$ are uniformly at random in $\mathcal{B}_n(R)$, the elements of the output set $\mathbf{S}'$ are uniformly at random in $\mathcal{B}_n(\gamma R)$.

- There exists a number $P_\beta(\gamma)$ such that for a random $(\beta-1)$-tuple $(\mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1}) \in \mathbf{C}^{\beta-1}$ and a random element $\mathbf{v}$ in $\mathbf{S}$, the probability that $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1})\big) \leq \gamma R$ is lower bounded by $P_\beta(\gamma)$.

- For random $\mathbf{v} \in \mathbf{S}$ and $(\mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1}) \neq (\mathbf{c}'_1, \ldots, \mathbf{c}'_{\beta-1}) \in \mathbf{C}^{\beta-1}$, the events $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1})\big) \leq \gamma R$ and $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}'_1, \ldots, \mathbf{c}'_{\beta-1})\big) \leq \gamma R$ are independent.

The goal is to derive a bound $N_{\mathbf{C}}$ such that if $|\mathbf{C}| \geq N_{\mathbf{C}}$, then with overwhelming probability for all $\mathbf{v} \in \mathbf{S} \backslash \mathbf{C}$ there exist $\mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1} \in \mathbf{C}^{i-1}$ such that $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1})\big) \leq \gamma R$. As a consequence, $|\mathbf{C}|$ will with overwhelming probability not exceed $N_{\mathbf{C}}$ and the cardinality of the output set of Algorithm 10 can be bounded as follows:

$$|\mathbf{S}'| = |\mathbf{S}| - |\mathbf{C}| \geq |\mathbf{S}| - N_{\mathbf{C}}.$$

We prove a generalized version of Lemma 4.2.1.

Lemma 4.2.2 *Let $\beta, n \in \mathbb{N}$, $\gamma \in (0,1)$ and $\mathbf{S}$ a set of $N_{\mathbf{S}}$ points chosen independently uniformly at random from $\mathcal{C}_n(\gamma, R)$. Further let*

$$N_{\mathbf{C}} = (\beta - 1)\left(\frac{n}{P_\beta(\gamma)}\right)^{1/(\beta-1)},$$

and $P_\beta(\gamma) \in [0,1]$ as above. Let $N_{\mathbf{C}} < N_{\mathbf{S}} < 2^n$ and $\mathbf{C}$ be a random subset of $\mathbf{S}$ of cardinality at least $N_{\mathbf{C}}$. Then, with overwhelming probability for all $\mathbf{v} \in \mathbf{S}$ there exists a $(i-1)$-tuple $(\mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1}) \in \mathbf{C}^{\beta-1}$ such that $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1})\big) \leq \gamma R$.

Proof: Let $E_\beta(\gamma)$ denote the expected number of elements $\mathbf{v}$ in $\mathbf{S}\backslash\mathbf{C}$ such that there is no $(\beta-1)$-tuple $(\mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1}) \in \mathbf{C}^{\beta-1}$ with $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1})\big) \leq \gamma R$. Using the assumptions stated above we have

$$E_\beta(\gamma) \leq |\mathbf{S}\backslash\mathbf{C}| \cdot \big(1 - P_\beta(\gamma)\big)^{\binom{|\mathbf{C}|}{\beta-1}}.$$

If $E_\beta(\gamma) \leq 1$, then with overwhelming probability for all $\mathbf{v} \in \mathbf{S}\backslash\mathbf{C}$ there exists a $(\mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1}) \in \mathbf{C}^{\beta-1}$ with $\lambda_1\big(\mathcal{L}(\mathbf{v}, \mathbf{c}_1, \ldots, \mathbf{c}_{\beta-1})\big) \leq \gamma R$. It is sufficient to require

$$|\mathbf{S}\backslash\mathbf{C}| \cdot \big(1 - P_\beta(\gamma)\big)^{\binom{|\mathbf{C}|}{\beta-1}} \leq 1,$$

or equivalently

$$\binom{|\mathbf{C}|}{\beta-1} \log\big(1 - P_\beta(\gamma)\big) \leq -\log|\mathbf{S}\backslash\mathbf{C}|.$$

As $\log(1-x) \leq -x$ and $\binom{|\mathbf{C}|}{\beta-1} \geq \left(\frac{|\mathbf{C}|}{\beta-1}\right)^{\beta-1}$ we get the sufficient condition

$$\left(\frac{|\mathbf{C}|}{\beta-1}\right)^{\beta-1} \cdot P_\beta(\gamma) \geq \log|\mathbf{S}\backslash\mathbf{C}|,$$

which is equivalent to

$$|\mathbf{C}| \geq (\beta-1)\left(\frac{\log|\mathbf{S}\backslash\mathbf{C}|}{P_\beta(\gamma)}\right)^{1/(\beta-1)}.$$

□

Iteratively applying the sieving step as described in Algorithm 10 to a starting set $\mathbf{S}_0 \subset \mathcal{L} \cap \mathcal{B}_n(R_0)$ with $R_0 = 2^{\mathcal{O}(n)}\lambda_1(\mathcal{L})$ and cardinality $|\mathbf{S}_0| = \mathrm{poly}(n) N_{\mathbf{C}}$, where $N_{\mathbf{C}}$ is as in Lemma 4.2.2, we expect to find a shortest vector in the lattice. Unfortunately it is not clear to what extend the assumptions stated above hold and how to compute a good lower bound on the probability $P_i(\gamma)$.

Chapter 5

Measuring reducedness

The complexity of lattice problems highly depends on the quality, i.e. the reducedness of the given basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. Both the *orthogonality defect* $\mathrm{od}(B)$ and the *Seysen measure* $S(B)$ try to quantify the reducedness of the basis B. The orthogonality defect

$$\mathrm{od}(B) := \prod_{i=1}^{n} \frac{\|\mathbf{b}_i\|^2}{\|\mathbf{b}_i^*\|^2} = \frac{\prod_{i=1}^{n} \|\mathbf{b}_i\|^2}{\mathrm{vol}(\mathcal{L}(B))^2},$$

directly depends on the lengths of the basis vectors and can be seen as the canonical way to measure the reducedness of a basis. It is not hard to see that $\mathrm{od}(B) \geq 1$ with equality if and only if the basis vectors are pairwise orthogonal. The term orthogonality defect is not completely unambiguous, as the exact definitions vary in the literature. We use the same notion as e.g. [TMK05]. Sometimes the square is omitted e.g. in [MG02, Chapter 7] and [LJS90] or a normalized version $1 - \prod_{i=1}^{n} \frac{\|\mathbf{b}_i^*\|^2}{\|\mathbf{b}_i\|^2}$ to be in $[0, 1)$ is used e.g. in [Maz10] or, under the name orthogonality deficiency, in [ZAM08]. Low orthogonality defect trivially implies the existence of a relatively short basis vector:

$$\min_{1 \leq i \leq n} \|\mathbf{b}_i\|^2 \leq \left(\prod_{i=1}^{n} \|\mathbf{b}_i\|^2 \right)^{1/n} = \mathrm{od}(B)^{1/n} \mathrm{vol}(\mathcal{L})^{1/n},$$

and the following lemma from [Lod09, Thm 14.13] shows the connection between the orthogonality defect and the size of the coefficients of a shortest vector.

Lemma 5.0.3 *Given a lattice basis B with orthogonality defect $\mathrm{od}(B)$. Then a shortest nonzero vector $\mathbf{v} \in \mathcal{L}(B)$ is of the form*

$$\mathbf{v} = \sum_{i=1}^{n} v_i \mathbf{b}_i \quad \textit{with} \quad |v_i| \leq \sqrt{\mathrm{od}(B)}.$$

The orthogonality defect of a HKZ-basis can be upper bounded by $\gamma_n^n \prod_{i=1}^{n} \frac{i+3}{4}$ [LJS90], where γ_n as usual denotes the n-th Hermite constant. This proves the existence of a basis B with $\mathrm{od}(B) \leq \exp(2n \ln n)$.

The Seysen measure

$$S(B) := \sum_{i=1}^{n} \|\mathbf{b}_i\|^2 \|\mathbf{b}_{n-i+1}^{\times}\|^2,$$

depends on both, the lengths of the basis vectors of B and of its reverse dual basis $B^{\times} = [\mathbf{b}_1^{\times}, \ldots, \mathbf{b}_n^{\times}]$. It holds that $S(B) \geq n$ with equality if and only if the basis vectors are pairwise orthogonal. The Seysen measure was introduced by Seysen [Sey93] as part of a lattice basis reduction algorithm. The algorithm reduces the Seysen measure of a basis by iteratively performing elementary row operations on the basis matrix B. For a very comprehensive description and discussion of the algorithm we refer to [LaM91]. It is known [Sey93] that for every lattice there exists a basis such that $S(B) \leq \exp(\mathcal{O}\left(\ln(n)^2\right))$ and recently Maze [Maz10] made this bound precise by showing that $S(B) \leq \exp((2/\ln 2 + 1)\ln(n)^2 + 4\ln n)$. Remarkably enough, the proofs of these bounds again rely on the existence of a HKZ-basis. A similar result as the one in Lemma 5.0.3 can also be shown by means of the Seysen measure: If two points in $\mathrm{span}(\mathcal{L}(B))$ are close to each other, the distance of its coefficients with respect to B can be bounded. More precisely, for two points $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n$ and $\mathbf{w} = w_1\mathbf{b}_1 + \cdots + w_n\mathbf{b}_n$ in $\mathrm{span}(\mathcal{L})$ it can be shown (see Lemma 5.1.4) that

$$\|(v_1 - w_1, \ldots, v_n - w_n)\|^2 \leq \|\mathbf{v} - \mathbf{w}\|^2 \frac{S(B)}{\lambda_1(\mathcal{L})^2}. \tag{5.0.1}$$

Several inequalities relating the Seysen measure of a basis to the orthogonality defect exist. Zhang et al. [ZAM08] showed that

$$\frac{1}{n} S(B) \leq \mathrm{od}(B) \leq \frac{1}{n} \left(S(B) - n + 1\right)^{n-1}. \tag{5.0.2}$$

The first inequality is a direct consequence of a result by Taherzadeh et al. [TMK05, Proof of Lemma 1] saying that $\|\mathbf{b}_i\|\|\mathbf{b}_i^{\times}\| \leq \sqrt{\mathrm{od}(B)}$, $1 \leq i \leq n$.

The motivational result of this chapter is due to Maze [Maz10] and improves the second inequality in (5.0.2):

$$\mathrm{od}(B) \leq \min\left\{ e\left(\frac{S(B)+1}{n}\right)^{n-1}, \left(\frac{S(B)}{n}\right)^{n} \right\}. \tag{5.0.3}$$

While we will see that the inequality $\mathrm{od}(B) \leq \left(\frac{S(B)}{n}\right)^n$ is a direct consequence of the geometric-arithmetic mean inequality, the proof of $\mathrm{od}(B) \leq e\left(\frac{S(B)+1}{n}\right)^{n-1}$ is a combination of two results. The first result is due to Maze [Maz10], who realized that the Seysen measure equals the trace of a symmetric positive definite matrix, while the orthogonality defect equals the determinant thereof. As such the Seysen measure equals the sum of the corresponding eigenvalues and the orthogonality defect the product thereof. Further the harmonic mean of the eigenvalues equals 1. The second result follows directly from our work published in [MW12], which implies the following lemma

Lemma 5.0.4 *For strictly positive reals $x_1, \ldots, x_n$ with harmonic mean $\frac{n}{\sum_{i=1}^n x_i^{-1}} = 1$ the following inequality holds:*

$$\prod_{i=1}^{n} x_i \leq e \left(\frac{\sum_{i=1}^n x_i + 1}{n} \right)^{n-1}.$$

The rest of the chapter is organized as follows. In Section 5.1 we review the two measures and their connections and show how Equation (5.0.3) can be derived using Lemma 5.0.4. We further prove Equation (5.0.1) (Lemma 5.1.4). In Section 5.2 the result leading to Lemma 5.0.4 is presented. We show in full detail how the inequality between the harmonic and arithmetic mean, the arithmetic and geometric mean respectively, can be tightened if the geometric mean, the harmonic mean respectively, is given.

5.1 Seysen measure vs. orthogonality defect

Let $\mathcal{L} \subset \mathbb{R}^m$ be a lattice of rank n with basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. Let as usual $B^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ denote its Gram-Schmidt orthogonalized basis and $B^\times = [\mathbf{b}_1^\times, \ldots, \mathbf{b}_n^\times]$ its reverse dual basis.

The i-th Gram-Schmidt vector $\mathbf{b}_i^*$ is orthogonal to $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$ and $\mathbf{b}_i - \mathbf{b}_i^*$ is the orthogonal projection of $\mathbf{b}_i$ onto $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$. By $\varphi_i \in [0, \pi]$ we denote the angle between $\mathbf{b}_i$ and $\mathbf{b}_i - \mathbf{b}_i^*$ (compare Figure 1a). We have

$$\frac{\|\mathbf{b}_i\|}{\|\mathbf{b}_i^*\|} = \sin(\varphi_i)^{-1}.$$

As an immediate consequence from the definition we have that $\mathbf{b}_{n-i+1}^\times$ is orthogonal to $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n)$. We define $\psi_i \in [0, \pi]$ to be the angle between $\mathbf{b}_i$ and its orthogonal projection onto $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n)$ (compare Figure 1b). From $1 = \langle \mathbf{b}_i, \mathbf{b}_{n-i+1}^\times \rangle = \|\mathbf{b}_i\| \|\mathbf{b}_{n-i+1}^\times\| \cos(\pi/2 - \psi_i) = \|\mathbf{b}_i\| \|\mathbf{b}_{n-i+1}^\times\| \sin(\psi_i)$ it follows that

$$\|\mathbf{b}_i\| \|\mathbf{b}_{n-i+1}^\times\| = \sin(\psi_i)^{-1}.$$

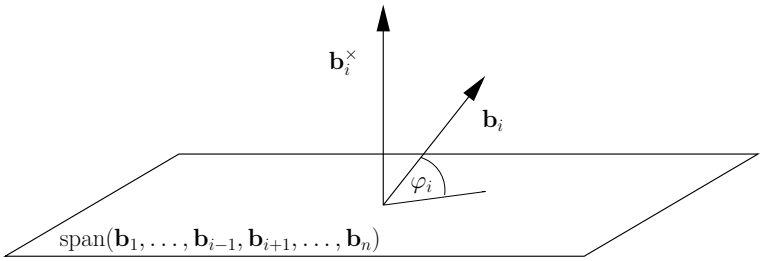
Consequently we have the following alternative expressions for Seysen measure and the orthogonality defect

$$\operatorname{od}(B) = \prod_{i=1}^{n} \frac{\|\mathbf{b}_i\|^2}{\|\mathbf{b}_i^*\|^2} = \prod_{i=1}^{n} \frac{1}{\sin^2 \varphi_i}, \tag{5.1.4}$$

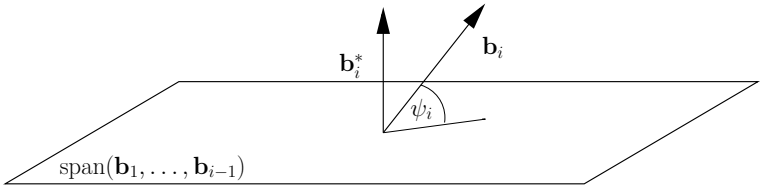
$$S(B) = \sum_{i=1}^{n} \|\mathbf{b}_i\|^2 \|\mathbf{b}_{n-i+1}^\times\|^2 = \sum_{i=1}^{n} \frac{1}{\sin^2 \psi_i}. \tag{5.1.5}$$

With $\psi_i \leq \varphi_i$ and the geometric-arithmetic mean inequality we get

$$\frac{1}{n} S(B) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sin^2 \psi_i} \geq \left(\prod_{i=1}^{n} \frac{1}{\sin^2 \psi_i} \right)^{1/n} \geq \left(\prod_{i} \frac{1}{\sin^2 \varphi_i} \right)^{1/n} = \operatorname{od}(B)^{1/n},$$



(a) Angle between $\mathbf{b}_i$ and span of remaining basis vectors.



(b) Angle between $\mathbf{b}_i$ and span of preceding basis vectors.

Figure 1

showing the first part of Equation (5.0.3):

$$\mathrm{od}(B) \leq \left(\frac{S(B)}{n}\right)^n. \tag{5.1.6}$$

For the second part consider B^TB. As the Gram-matrix of a set of linearly independent vectors in $\mathbb{R}^m$ it is symmetric positive definite. Writing $B = \mathrm{diag}(\|\mathbf{b}_1\|, \ldots, \|\mathbf{b}_n\|) \cdot \tilde{B}$, we have $B^TB = \mathrm{diag}(\|\mathbf{b}_1\|, \ldots, \|\mathbf{b}_n\|) \cdot \tilde{B}^T\tilde{B} \cdot \mathrm{diag}(\|\mathbf{b}_1\|, \ldots, \|\mathbf{b}_n\|)$ and hence $M := \tilde{B}^T\tilde{B} \in \mathbb{R}^{n\times n}$ is symmetric positive definite. As such also M^{-1} is symmetric positive definite and has positive real eigenvalues $\xi_1, \ldots, \xi_n \in \mathbb{R}$. It is possible to express both the Seysen measure and the orthogonality defect by means of these eigenvalues, as has been shown by [Maz10, Section 3]. We summarize this result in the following proposition:

Proposition 5.1.1 *Using the notation from above, let $\xi_1, \ldots, \xi_n \in \mathbb{R}$ be the eigenvalues of the matrix $M^{-1} = (\tilde{B}^T\tilde{B})^{-1}$. Then the following holds:*

$$\mathrm{od}(B) = \det\left(M^{-1}\right) = \prod_{i=1}^{n} \xi_i,$$

$$S(B) = \mathrm{trace}\left(M^{-1}\right) = \sum_{i=1}^{n} \xi_i.$$

Further

$$\mathrm{trace}\,(M) = \sum_{i=1}^{n} \frac{1}{\xi_i} = n.$$

Note the similarity to Equation (5.1.4) and (5.1.5). However the eigenvalues ξ_i are not equal $\sin(\varphi_i)^{-1}$, $\sin(\psi_i)^{-1}$ respectively, in general. In fact we have equality if and only if the basis vectors are pairwise orthogonal.

The usual harmonic-geometric-arithmetic mean inequalities imply the following inequalities

$$\underbrace{\frac{n}{\sum_{i=1}^{n}\xi_i^{-1}}}_{=1} \leq \underbrace{\left(\prod_{i=1}^{n}\xi_i\right)^{1/n}}_{=\mathrm{od}(B)^{1/n}} \leq \underbrace{\frac{1}{n}\sum_{i=1}^{n}\xi_i}_{=\frac{1}{n}S(B)}. \tag{5.1.7}$$

Note that these inequalities are true in general, i.e. for arbitrary positive reals ξ_i, and that the second inequality again implies (5.1.6) However a resent result [MW12] allows to make the second inequality more exact, using the fact the concrete value of the harmonic mean is given.

Lemma 5.1.2 *Let $x_1, \ldots, x_n$ be strictly positive reals with harmonic mean $\frac{n}{\sum_{i=1}^{n} x_i^{-1}} = 1$. Then*

$$\prod_{i=1}^{n} x_i \leq e\left(\frac{\sum_{i=1}^{n} x_i + 1}{n}\right)^{n-1}.$$

Applying the lemma to the situation as summarized in Equation (5.1.7) we get that

$$\mathrm{od}(B) = \prod_{i=1}^{n}\xi_i \leq e\left(\frac{\sum_{i=1}^{n}\xi_i + 1}{n}\right)^{n-1} = e\left(\frac{S(B)+1}{n}\right)^{n-1}.$$

Hence

Corollary 5.1.3 *Using the notation above the following inequality holds*

$$\mathrm{od}(B)^2 \leq \min\left\{e\left(\frac{S(B)+1}{n}\right)^{n-1}, \left(\frac{S(B)}{n}\right)^{n}\right\}.$$

In Lemma 5.0.3 we have seen that the coefficients of a short vector with respect to a basis B can be bounded by the orthogonality defect. The following lemma provides a similar result with respect to the Seysen measure of a basis. More concretely it shows the connection between the distance of two points in $\mathbb{R}^m$ and the distance of the respective coefficient vectors with respect to a basis B.

Lemma 5.1.4 *Let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be a basis of a lattice $\mathcal{L} \in \mathbb{R}^m$ and let $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n$ and $\mathbf{w} = w_1\mathbf{b}_1 + \cdots + w_n\mathbf{b}_n$ be two points in $\mathrm{span}(\mathcal{L})$, with coefficient vectors $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$ and $w = (w_1, \ldots, w_n) \in \mathbb{R}^n$. Then*

$$\|v - w\|^2 \leq \|\mathbf{v} - \mathbf{w}\|^2 \frac{S(B)}{\lambda_1(\mathcal{L})^2}.$$

Proof: Let $\|\cdot\|_F$ denote the well known Frobenius norm, i.e.

$$\|(a_{i,j})_{1\leq i,j\leq n}\|_F = \left(\sum_{1\leq i,j\leq n} |a_{i,j}|^2\right)^{1/2}.$$

By the submultiplicity we have

$$\|v-w\|_F^2 = \|vBB^{-1} - wBB^{-1}\|_F^2 \leq \|vB - wB\|_F^2 \|B^{-1}\|_F^2 = \|\mathbf{v}-\mathbf{w}\|_F^2 \|B^{-1}\|_F^2.$$

Clearly $\|B^{-1}\|_F^2 = \sum_{i=1}^n \|\mathbf{b}_i^\times\|^2$. As $\lambda_1(\mathcal{L}) \leq \|\mathbf{b}_i\|$ for $i = 1,\ldots,n$ we have

$$\|B^{-1}\|_F^2 = \sum_{i=1}^n \|\mathbf{b}_i^\times\|^2 \leq \frac{1}{\lambda_1(\mathcal{L})^2} \sum_{i=1}^n \|\mathbf{b}_i\|^2 \|\mathbf{b}_i^\times\|^2 = \frac{S(B)}{\lambda_1(\mathcal{L})^2}$$

and the claim follows. □

Corollary 5.1.5 *Given a lattice basis B with Seysen measure $S(B)$. Then a shortest nonzero vector $\mathbf{v} \in \mathcal{L}(B)$ is of the form*

$$\mathbf{v} = \sum_{i=1}^n v_i \mathbf{b}_i \quad \textit{with} \quad \|(v_1,\ldots,v_n)\| \leq \sqrt{S(B)}.$$

5.2 Harmonic-geometric-arithmetic means inequalities

In this section we review the results on harmonic, geometric and arithmetic means inequalities obtained in close collaboration with G. Maze and published in [MW12].

Definition 5.2.1 *Given strictly positive reals $x_1,\ldots,x_n \in \mathbb{R}_{>0}$ and weights $\alpha_1,\ldots,\alpha_n \in \mathbb{R}_{>0}$ the* weighted harmonic mean *is defined as*

$$H(x,\alpha) := \left(\frac{1}{\sum_{i=1}^n \alpha_i} \sum_{i=1}^n \frac{\alpha_i}{x_i}\right)^{-1},$$

the weighted geometric mean *is defined as*

$$G(x,\alpha) := \prod_{i=1}^n x_i^{\frac{\alpha_i}{\sum_{i=1}^n \alpha_i}},$$

and the weighted arithmetic mean *is defined as*

$$A(x,\alpha) := \frac{1}{\sum_{i=1}^n \alpha_i} \sum_{i=1}^n \alpha_i x_i.$$

Multiplying the weight vector α by a positive real $\lambda \in \mathbb{R}_{>0}$ does not change the three means. In the sequel we will therefore assume that the weights are normalized, i.e. that $\sum_{i=1}^{n} \alpha_i = 1$. Also we will often skip the 'weighted' when talking about the three means. The well known harmonic-geometric-arithmetic mean inequalities are given by

$$H(x,\alpha) \leq G(x,\alpha) \leq A(x,\alpha),$$

and both inequalities reach equality if and only if $x_i = x_j$, $1 \leq i,j \leq n$.

In this section we derive bounds on the harmonic mean, geometric mean respectively, when the arithmetic and geometric mean, arithmetic and harmonic mean respectively, are given. More concretely, for $\alpha_1,\ldots,\alpha_n \in \mathbb{R}_{>0}$ such that $\sum_{i=1}^{n} \alpha_i = 1$ and $G,H,A \in \mathbb{R}_{>0}$ with $G,H < A$, we upper and lower bound $H(x,\alpha)$ for all $x \in \mathbb{R}^n_{>0}$ such that $A(x,\alpha) = A$ and $G(x,\alpha) = G$ (Theorem 5.2.7). Similarly we upper and lower bound $G(x,\alpha)$ for all $x \in \mathbb{R}^n_{>0}$ such that $A(x,\alpha) = A$ and $H(x,\alpha) = H$ (Theorem 5.2.8).

The derived bounds depend on the smallest weight $\min_{i=1,\ldots,n} \alpha_i$ and the given means only and the vectors $x \in \mathbb{R}^n_{>0}$ reaching these bounds have the following special structure

$$x = (a,\ldots,a,b,a,\ldots,a), \quad \text{for some} \quad a,b \in \mathbb{R}_{>0}. \tag{5.2.8}$$

Their coefficients are all equal exept for one corresponding to a minimal weight, i.e. $x_i = a$ for some i such that $\alpha_i = \min_{i=1,\ldots,n} \alpha_i$ and $x_j = b$ for $1 \leq j \leq n$ with $j \neq i$.

We proceed as follows. In Section 5.2.1 we will discuss the two-dimensional case. We will show that if $x = (x_1,x_2) \in \mathbb{R}^2_{>0}$ has given arithmetic and geometric mean, its harmonic mean can take at most two different values (Lemma 5.2.3). Similarly if the arithmetic and harmonic mean are given, the geometric mean can take at most two values (Lemma 5.2.5). These values depend on the given means and the weight vector α only and we will see how they change depending on α (Lemma 5.2.4, respectively Lemma 5.2.6).

In Section 5.2.2 we will see, using Lagrange multipliers, that if $x \in \mathbb{R}^n_{>0}$ is maximizing/minimizing the harmonic mean under the constraints that $A(x,\alpha) = A$ and $G(x,\alpha) = G$, then x has the special structure as in Equation (5.2.8). Similarly if it is maximizing/minimizing the geometric mean under the constraints that $A(x,\alpha) = A$ and $H(x,\alpha) = H$ then x has also the special structure as in Equation (5.2.8). The means of vectors of this special form can be seen as two-dimensional means of the two values a and b only, where the weight vector is changed accordingly. So the considerations in the two-dimensional case from Section 5.2.2 will lead to the main results (Theorem 5.2.7, respectively Theorem 5.2.8). A corollary directly implying Lemma 5.0.4 is given (Corollary 5.2.10).

Note that for $\lambda \in \mathbb{R}_{>0}$, we have that $A(\lambda x,\alpha) = \lambda A(x,\alpha)$ and the same holds for the other two means. We will therefore often assume that $A = 1$ and $G,H < 1$.

5.2.1 Two dimensional case

We start with the case where the arithmetic and geometric mean are fixed and the harmonic mean is to be bounded. The case where the arithmetic and harmonic mean

are fixed and bounds on the geometric mean are derived follows.

Bounds on the harmonic mean

Assume the (weighted) arithmetic and geometric mean of two positive reals x_1, x_2 with normalized weights $\alpha_1, \alpha_2 \in \mathbb{R}_{>0}$ are given:

$$A(x, \alpha) = 1, \tag{5.2.9}$$
$$G(x, \alpha) = G. \tag{5.2.10}$$

We would like to have an upper and lower bound on the corresponding harmonic mean:

$$H(x, \alpha) = \left(\frac{\alpha_1}{x_1} + \frac{\alpha_2}{x_2} \right)^{-1},$$

i.e. we would like to maximize and minimize $H(x, \alpha)$ dependent on $x_1, x_2, \alpha_1, \alpha_2$ under the conditions given in (5.2.9), (5.2.10) and by $\alpha_1 + \alpha_2 = 1$. As already mentioned, $A(x, \alpha) = G(x, \alpha)$ if and only if $x_1 = x_2$ and in this case $H(x, \alpha) = A(x, \alpha) = G(x, \alpha)$. So the interesting case is when $G(x, \alpha) = G < 1$. With the conditions $\alpha_1 + \alpha_2 = 1$ and $A(x, \alpha) = 1$ the number of variables can be reduced from four to two. Recalling that $\alpha_1, \alpha_2, x_1, x_2 \in \mathbb{R}_{>0}$ we have the following equivalence:

$$\left.\begin{array}{l} \alpha_1 + \alpha_2 = 1 \\ \alpha_1 x_1 + \alpha_2 x_2 = 1 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{ll} \alpha_i = 1 - \alpha_j & \\ x_i = \frac{1-\alpha_j x_j}{1-\alpha_j} & i, j \in \{1, 2\}, \quad j \neq i. \end{array}\right.$$

Without loss of generality let $\alpha_1 \leq 1/2$. The harmonic and geometric mean can then be written as functions of x_1 and α_1 only,

$$\begin{array}{rcll} h(x_1, \alpha_1) & := & \frac{x_1(1-\alpha_1 x_1)}{x_1(1-2\alpha_1)+\alpha_1} & = \quad H(x, \alpha), \\ g(x_1, \alpha_1) & := & x_1^{\alpha_1} \left(\frac{1-\alpha_1 x_1}{1-\alpha_1} \right)^{1-\alpha_1} & = \quad G(x, \alpha), \end{array}$$

This allows to reformulate the problem as follows: Find $0 < \alpha_1 \leq \frac{1}{2}$ and $0 \leq x_1 < \frac{1}{\alpha_1}$ satisfying $g(x_1, \alpha_1) = G$ such that $h(x_1, \alpha_1)$, is minimal, maximal respectively. We will first solve the problem for fixed $0 < \alpha_1 \leq \frac{1}{2}$. Consider the functions

$$\begin{array}{rccl} h_{\alpha_1}: & [0, 1/\alpha_1] & \to & \mathbb{R} \\ & y & \mapsto & h(y, \alpha_1) = \frac{y(1-\alpha_1 y)}{y(1-2\alpha_1)+\alpha_1} \end{array}$$

and

$$\begin{array}{rccl} g_{\alpha_1}: & [0, 1/\alpha_1] & \to & \mathbb{R} \\ & y & \mapsto & g(y, \alpha_1) = y^{\alpha_1} \left(\frac{1-\alpha_1 y}{1-\alpha_1} \right)^{1-\alpha_1} \end{array}$$

The following lemma describes the behaviour of the functions g_{α_1} and $\sqrt{h_{\alpha_1}}$ (compare also Figure 2).

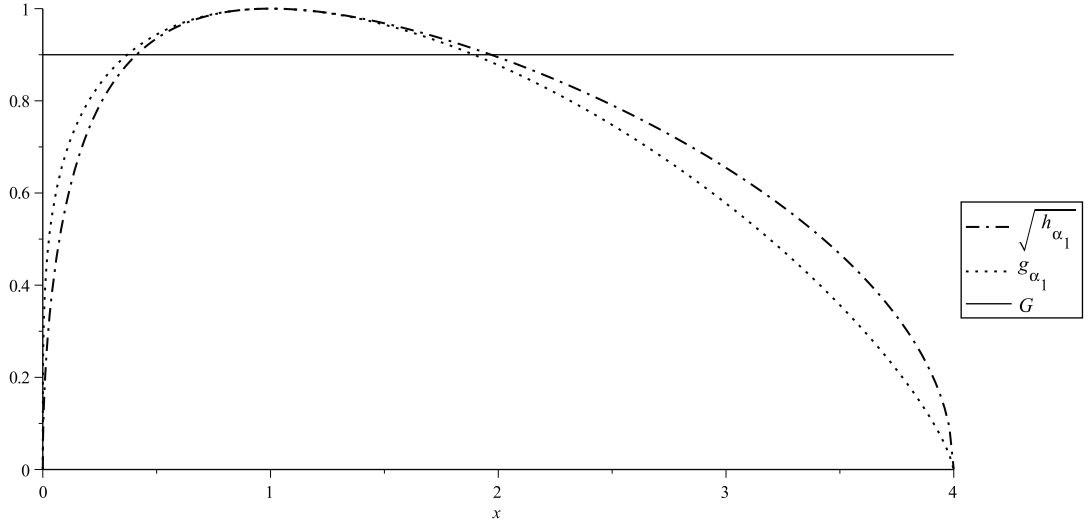


Figure 2: f_{α_1} and $\sqrt{h_{\alpha_1}}$ for $\alpha_1 = 1/4$. $G = 0.9$.

Lemma 5.2.2 *Using the notation above, we have that* $\sqrt{h_{\alpha_1}}$ *and* g_{α_1} *are strictly increasing in* $[0,1]$*, and strictly decreasing in* $[1, 1/\alpha_1]$*. Further*

$$0 = g_{\alpha_1}(y) = \sqrt{h_{\alpha_1}(y)} \quad \textit{for} \quad y \in \{0, \alpha_1^{-1}\}, \tag{5.2.11}$$

$$1 = g_{\alpha_1}(y) = \sqrt{h_{\alpha_1}(y)} \quad \textit{for} \quad y = 1, \tag{5.2.12}$$

$$1 > g_{\alpha_1}(y) \geq \sqrt{h_{\alpha_1}(y)} > 0 \quad \textit{for} \quad y \in (0,1), \tag{5.2.13}$$

$$1 > \sqrt{h_{\alpha_1}(y)} \geq g_{\alpha_1}(y) > 0 \quad \textit{for} \quad y \in (1, \alpha_1^{-1}). \tag{5.2.14}$$

The Inequalities (5.2.13) *and* (5.2.14) *reach equality if and only if* $\alpha_1 = 1/2$.

Proof: Equations (5.2.11) and (5.2.12) are immediate. For $y \in (0, 1/\alpha_1)$ the derivatives of $g_{\alpha_1}(y)$, $\sqrt{h_{\alpha_1}}(y)$ respectively are given by

$$\frac{\partial}{\partial y} g_{\alpha_1}(y) = \frac{\alpha_1 y^{\alpha_1}(y^{-1} - 1)}{1 - \alpha_1} \left(\frac{1 - \alpha_1}{1 - \alpha_1 y} \right)^{\alpha_1},$$

respectively

$$\frac{\partial}{\partial y} \sqrt{h_{\alpha_1}}(y) = \frac{1}{2} \left(\frac{y + \alpha_1 - 2\alpha_1 y}{y - \alpha_1 y^2} \right)^{1/2} \frac{\alpha_1(1 - 2\alpha_1 y + y^2(2\alpha_1 - 1))}{(2\alpha_1 y - y - \alpha_1)^2}.$$

One readily verifies that the two derivatives equal 0 for $y = 1$ and are strictly positive for $y \in (0,1)$, and strictly negative for $y \in (1, 1/\alpha_1)$. It remains to show that $g_{\alpha_1}(y) \geq \sqrt{h_{\alpha_1}(y)}$ for $y \in (0,1)$ and $g_{\alpha_1}(y) \leq \sqrt{h_{\alpha_1}(y)}$ for $y \in (1, 1/\alpha_1)$ with equality if and only if $\alpha_1 = 1/2$. It is immediate to see that the function coincide if $\alpha_1 = 1/2$. So assume

that $\alpha_1 < 1/2$. Consider the quotient $\frac{g_{\alpha_1}}{\sqrt{h_{\alpha_1}}}(y)$ defined for $y \in (0, 1/\alpha_1)$. Its derivative equals

$$\frac{\partial}{\partial y}\frac{g_{\alpha_1}}{\sqrt{h_{\alpha_1}}} = \alpha_1 y^{\alpha_1 - 1}\frac{1}{2}\left(\frac{\alpha_1 - 1}{\alpha_1 y - 1}\right)^{\alpha_1}\left(\frac{y(2\alpha_1 - 1) - \alpha_1}{y(\alpha_1 y - 1)}\right)^{1/2}\frac{(2\alpha_1 - 1)(x-1)^2}{(\alpha - 1)(y(2\alpha_1 - 1) - \alpha_1)}.$$

By inspection one can see that this is strictly negative for $y \in (0, 1/\alpha_1)$. As $\frac{g_{\alpha_1}}{\sqrt{h_{\alpha_1}}}(1) = 1$, it follows that $g_{\alpha_1} > \sqrt{h_{\alpha_1}}$ on the interval $(0, 1)$ and $g_{\alpha_1} < \sqrt{h_{\alpha_1}}$ on the interval $(1, 1/\alpha_1)$. □

As $g_{\alpha_1}(y)$ is continuous on $\left[0, \alpha_1^{-1}\right]$ and $G < 1$ we can conclude that the equation

$$g_{\alpha_1}(y) = y^{\alpha_1}\left(\frac{1 - \alpha_1 y}{1 - \alpha_1}\right)^{1-\alpha_1} = G, \tag{5.2.15}$$

has exactly 2 solutions x_1, x_1' with $0 < x_1' < 1 < x_1 < 1/\alpha_1$ in the interval $\left[0, \alpha_1^{-1}\right]$ (compare Figure 2). Further we have that

$$\sqrt{h_{\alpha_1}(x_1')} \leq g_{\alpha_1}(x_1') = G = g_{\alpha_1}(x_1) \leq \sqrt{h_{\alpha_1}(x_1)}.$$

So we have the following lemma

Lemma 5.2.3 *Let $0 < \alpha_1 \leq \frac{1}{2}$, $\alpha_2 = 1 - \alpha_1$ and $0 < G < 1$. Write $\alpha = (\alpha_1, \alpha_2)$. There are exactly two pairs of real positive numbers $x = (x_1, x_2)$ and $x' = (x_1', x_2')$ where $x_1' < 1 < x_1$ such that*

$$A(x, \alpha) = A(x', \alpha) = 1,$$
$$G(x, \alpha) = G(x, \alpha) = G.$$

Further x_1 and x_1' are the unique solutions of Equation (5.2.15) in the interval $\left(0, \alpha_1^{-1}\right)$ and

$$G > H(x, \alpha) = h_{\alpha_1}(x_1) \geq G^2 \geq h_{\alpha_1}(x_1') = H(x', \alpha) > 0,$$

reaching equality if and only if $\alpha_1 = 1/2$. In that case $H(x, \alpha) = G(x, \alpha)^2 = G^2$.

From Lemma 5.2.3 we see that for a fixed arithmetic and fixed smaller geometric mean, the harmonic mean can take at most two different values that depend only on the weight $0 < \alpha_1 \leq 1/2$. Using the notation of Lemma 5.2.3 let us write $\overline{H}_G(\alpha_1) := H(x, \alpha)$ and $\underline{H}_G(\alpha_1) := H(x', \alpha)$ respectively. For all $0 < \alpha_1 \leq 1/2$ and $G < 1$, it holds that $\overline{H}_G(\alpha_1) \geq \underline{H}_G(\alpha_1)$ and as a consequence

$$\sup_{0<\alpha_1\leq 1/2} \overline{H}_G(\alpha_1),$$
$$\inf_{0<\alpha_1\leq 1/2} \underline{H}_G(\alpha_1),$$

give an upper, lower bound respectively on the weighted harmonic mean of two values having normalized weighted arithmetic mean and fixed weighted geometric mean G. The question that remains to investigate is, for which value $0 < \alpha_1 \leq \frac{1}{2}$, we have that $\overline{H}_G(\alpha_1)$ gets maximal, $\underline{H}(\alpha_1)$ gets minimal respectively. Table 1 suggests that $\underline{H}_G$ is increasing in α_1 and $\overline{H}_G$ is decreasing in α_1. In fact this is the case:

A	G	α_1	x_1'	x_1	$\underline{H}_G(\alpha_1) = h_{\alpha_1}(x_1')$	$\underline{H}_G(\alpha_1) = h_{\alpha_1}(x_1)$
1	0.8	0.1	0.043	3.97	0.320	0.730
1	0.8	0.2	0.152	2.62	0.506	0.703
1	0.8	0.3	0.248	2.10	0.574	0.682
1	0.8	0.4	0.329	1.80	0.613	0.662
1	0.8	0.5	0.400	1.60	0.640	0.640

Table 1: Example for the possible harmonic means of two values having normalized arithmetic mean $A = 1$ and geometric mean $G = 0.8$. The value in the table are rounded.

Lemma 5.2.4 *For any $G \in (0,1)$, $\overline{H}_G(\alpha_1)$ is decreasing over $(0, 1/2]$ and $\underline{H}_G(\alpha_1)$ is increasing over $(0, 1/2]$.*

Proof: Let $x_1' < 1 < x_1$ denote the solutions of Equation (5.2.15) as above. Note that Equation (5.2.15) gives us an implicit function

$$g(y, \alpha_1) = g_{\alpha_1}(y) = y^{\alpha_1} \left(\frac{1 - \alpha_1 y}{1 - \alpha_1} \right)^{1-\alpha_1} = G.$$

As for $G < 1$, $\left(\frac{\partial}{\partial y} g_{\alpha_1}\right)(x_1) \neq 0$ and $\left(\frac{\partial}{\partial y} g_{\alpha_1}\right)(x_1') \neq 0$ the implicit function theorem tells us that we can locally write the solutions x_1 and x_1' of the above equations as differentiable functions of α_1:

$$x_1' = x_1'(\alpha_1),$$
$$x_1 = x_1(\alpha_1).$$

Implicit differentiation allows us to find the derivatives of these function with respect to α_1. A somewhat lengthy computation shows [MW12, proof of Lemma 3.4] that

$$\frac{\partial}{\partial \alpha_1} \overline{H}_G(\alpha_1) = \frac{\partial}{\partial \alpha_1} H_G(\alpha_1, x_1(\alpha_1)) \leq 0,$$
$$\frac{\partial}{\partial \alpha_1} \underline{H}_G(\alpha_1) = \frac{\partial}{\partial \alpha_1} H_G(\alpha_1, x_1'(\alpha_1)) \geq 0,$$

which finishes the proof. □

Bounds on the geometric mean

Let us now consider the case where the arithmetic mean and the harmonic mean are fixed, and we want to upper and lower bound the geometric mean. Again we assume that we have normalized arithmetric mean $A(x, \alpha) = 1$ and

$$H(x, \alpha) = H < 1.$$

As in Section 5.2.1 we can write the geometric and harmonic mean as functions in two variables,

$$H(x, \alpha) = h(x_1, \alpha_1),$$
$$G(x, \alpha) = g(x_1, \alpha_1),$$

where we again assume without loss of generality that $0 < \alpha_1 \leq 1/2$. We want to maximize and minimize

$$g(x_1, \alpha_1) = x_1^{\alpha_1} \left(\frac{1 - \alpha_1 x_1}{1 - \alpha_1} \right)^{1-\alpha_1},$$

under the condition that

$$h(x_1, \alpha_1) = \frac{x_1(1 - \alpha_1 x_1)}{x_1(1 - 2\alpha_1) + \alpha_1} = H.$$

Again we start by fixing α_1 and considering the functions $g_{\alpha_1}(y)$ and $h_{\alpha_1}(y)$. Let $x_1' < 1 < x_1$ be the two unique solutions of

$$h_{\alpha_1}(y) = H. \tag{5.2.18}$$

By Lemma 5.2.2

$$g_{\alpha_1}(x_1') \geq \sqrt{h_\alpha(x_1')} = \sqrt{H} = \sqrt{h_\alpha(x_1)} \geq g_{\alpha_1}(x_1),$$

with equality if and only if $\alpha_1 = 1/2$. We directly get

Lemma 5.2.5 *Let $0 < \alpha_1 \leq 1/2$, $\alpha_2 = 1 - \alpha_1$ and $0 < H < 1$. Write $\alpha = (\alpha_1, \alpha_2)$. There are exactly two pairs of real positive numbers $x = (x_1, x_2)$ and $x' = (x_1', x_2')$ where $x_1' < 1 < x_1$ such that*

$$A(x, \alpha) = A(x', \alpha) = 1,$$
$$H(x, \alpha) = H(x', \alpha) = h_{\alpha_1}(x_1) = H.$$

Further x_1 and x_1' are the unique solutions of Equation (5.2.18) in the interval $(0, \alpha_1^{-1})$ and

$$H < G(x, \alpha) = g_{\alpha_1}(x_1) \leq \sqrt{H} \leq g_{\alpha_1}(x_1') = G(x', \alpha) < 1,$$

where equality is reached if and only if $\alpha_1 = 1/2$. In that case $G(x, \alpha) = \sqrt{H(x, \alpha)} = \sqrt{H}$.

Using the notation of the lemma, we define $\underline{G}_H(\alpha_1) := g_{\alpha_1}(x_1)$ and $\overline{G}_H(\alpha_1) := g_{\alpha_1}(x_1')$. Then

$$\inf_{0<\alpha_1\leq 1/2} \underline{G}_H(\alpha_1),$$
$$\sup_{0<\alpha_1\leq 1/2} \overline{G}_H(\alpha_1),$$

give a lower, respectively upper bound on the weighted geometric mean of two values having normalized weighted arithmetic mean and fixed weighted harmonic mean. The question that remains how these values behave depending on α_1. Table 2 gives an intuition the the behaviour.

A	H	α_1	x_1'	x_1	$\overline{G}_H(\alpha_1) = g_{\alpha_1}(x_1')$	$\underline{G}_H(\alpha_1) = g_{\alpha_1}(x_1)$
1	0.8	0.1	0.238	3.36	0.932	0.858
1	0.8	0.2	0.357	2.24	0.916	0.873
1	0.8	0.3	0.437	1.83	0.910	0.881
1	0.8	0.4	0.500	1.60	0.901	0.888
1	0.8	0.5	0.553	1.44	0.894	0.894

Table 2: Example for the possible geometric means of two values having normalized arithmetic mean $A = 1$ and harmonic mean $H = 0.8$. The values in the table are rounded.

Lemma 5.2.6 *For any $H \in [0, 1)$, $\overline{G}_H(\alpha_1)$ is decreasing over $(0, 1/2]$ and $\underline{G}_H(\alpha_1)$ is increasing over $(0, 1/2]$.*

Proof: Similar to the proof of Lemma 5.2.4, the implicit function

$$h_{\alpha_1}(y) = H,$$

locally gives us the two solutions $x_1' < 1 < x_1$ as differentiable functions of α_1:

$$x_1' = x_1'(\alpha_1),$$
$$x_1 = x_1(\alpha_1).$$

It can then be shown [MW12, proof of Lemma 3.6] that

$$\frac{\partial}{\partial\alpha_1}\overline{G}_H(\alpha_1) = \frac{\partial}{\partial\alpha_1}G_H(\alpha_1, x_1(\alpha_1)) \leq 0,$$
$$\frac{\partial}{\partial\alpha_1}\underline{G}_H(\alpha_1) = \frac{\partial}{\partial\alpha_1}G_H(\alpha_1, x_1'(\alpha_1)) \geq 0.$$

□

5.2.2 General case

Let us now consider the case where $n \geq 3$. For fixed normalized weights $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n_{>0}$ with $\sum_{i=1}^n \alpha_i = 1$ the (weighted) harmonic, geometric and arithmetric mean of $x \in \mathbb{R}^n_{>0}$ will be denoted by $H_\alpha(x)$, $G_\alpha(x)$ and $A_\alpha(x)$.

Bounds on the harmonic mean

For $x = (x_1, \ldots, x_n) \in \mathbb{R}^n_{>0}$ with given arithmetic mean $A_\alpha(x) = 1$ and geometric mean $G_\alpha(x) = G < 1$ we will upper and lower bound the corresponding harmonic mean:

$$H_\alpha(x) = \left(\sum_{i=1}^n \frac{\alpha_i}{x_i} \right)^{-1}.$$

We do this by upper respectively lower bounding the solutions to the following optimization problem.

$$\begin{aligned} \text{minimize/maximize} \quad & H_\alpha(x)^{-1} = \sum_{i=1}^n \frac{\alpha_i}{x_i}, \\ \text{under the contraints} \quad & A_\alpha(x) = \sum_{i=1}^n \alpha_i x_i = 1, \\ & \log(G_\alpha(x)) = \sum_{i=1}^n \alpha_i \log x_i = \log G, \\ & x \in X = \mathbb{R}^n_{>0} \subset \mathbb{R}^n. \end{aligned}$$

The reason for considering the inverse of the harmonic mean lies in the fact that this eases the computations. Clearly $X = \mathbb{R}^n_{>0} \subset \mathbb{R}^n$ is open and the gradients of the respective functions equal

$$\nabla H_\alpha^{-1} = \begin{pmatrix} \frac{-\alpha_1}{x_1^2} \\ \vdots \\ \frac{-\alpha_n}{x_n^2} \end{pmatrix}, \quad \nabla \log(G_\alpha) = \begin{pmatrix} \frac{\alpha_1}{x_1} \\ \vdots \\ \frac{\alpha_n}{x_n} \end{pmatrix} \quad \text{and} \quad \nabla A_\alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}.$$

The $H_\alpha(x)^{-1}$, $A_\alpha(x)$ and $\log(G_\alpha(x))$ are in $\mathcal{C}^1(X)$, i.e. they are continuously differentiable on X. Let

$$M := \{x \in X : \log(G_\alpha(x)) = \log G \wedge A_\alpha(x) = 1\}.$$

As $G < 1$ we have already seen that for $x \in M$ not all $x_i, i = 1, \ldots, n$ are equal. Consequently $\nabla A_\alpha(x)$ and $\nabla \log(G_\alpha)(x)$ are linearly independent for all $x \in M$. Further M is compact in X: As $\sum_{i=1}^n \alpha_i x_i = 1$ we have that $x_i \leq (\min_{j=1,\ldots,n} \alpha_j)^{-1}$. From $\sum_{i=1}^n \alpha_i \log x_i = \log G$ we get that

$$\log x_i = \alpha_i^{-1} \left(\log G - \sum_{j \neq i} \alpha_j \log x_j \right) \geq \alpha_i^{-1} \left(\log G + \log(\min_j \alpha_j) \sum_{j \neq i} \alpha_j \right).$$

Hence M is bounded and $M \cap X = M$. As we can write $M = \{x \in \mathbb{R}^n : A_\alpha(x) = 1\} \cap \{x \in \mathbb{R}^n_{>0} : \log(G_\alpha) = \log G\}$ and both sets being closed as preimages of continuous maps implies that M is compact. As a consequence, the supremum and infimum of H_α^{-1} on M are reached.

By e.g. [KÖ04, Section 3.6] it follows that if $x \in M$ is an extremal point for H_α^{-1} the following necessary condition is satisfied: There exist $\kappa_1, \kappa_2 \in \mathbb{R}$, called *Lagrange Multipliers*, such that

$$\nabla \left(H_\alpha(x)^{-1} - \kappa_1 A_\alpha(x) - \kappa_2 \log(G_\alpha(x)) \right) = 0.$$

So we have the necessary conditions

$$\frac{\alpha_i}{x_i^2} - \kappa_1 \alpha_i - \kappa_2 \frac{\alpha_i}{x_i} = 0, \quad 1 \leq i \leq n.$$

As $\alpha_i \neq 0$, equivalently

$$\frac{1}{x_i^2} - \kappa_1 - \kappa_2 \frac{1}{x_i} = 0, \quad 1 \leq i \leq n.$$

Note that this is a polynomial of degree 2, and as such it has at most two roots z_1 and z_2. So a necessary condition for the extremas is that for all $1 \leq i \leq n$, $x_i = z_1$ or $x_i = z_2$. From the condition that $G(x, \alpha) = G < 1$ we get that not all x_i's are the same, hence $z_1 \neq z_2$. Let $x \in M$ be a point satisfying the necessary consitions. Define $Z_1(x) = \{i : x_i = z_1\}$ and $Z_2(x) = \{i : x_i = z_2\}$. and

$$\begin{aligned} \boldsymbol{\alpha_1}(x) &:= \sum_{i \in Z_1(x)} \alpha_i, \\ \boldsymbol{\alpha_2}(x) &:= \sum_{i \in Z_2(x)} \alpha_i = 1 - \boldsymbol{\alpha}_1. \end{aligned}$$

Then the means can be written as functions of $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ only, i.e. we have that

$$\begin{aligned} A_\alpha(x) &= \sum_{i=1}^n \alpha_i x_i = \boldsymbol{\alpha_1} z_1 + \boldsymbol{\alpha_2} z_2 = 1, \\ G_\alpha(x) &= \prod_{i=1}^n x_i^{\alpha_i} = z_1^{\boldsymbol{\alpha_1}} z_2^{\boldsymbol{\alpha_2}} = G. \end{aligned}$$

and

$$H_\alpha(x) = \left(\sum_{i=1}^n \frac{\alpha_i}{x_i} \right)^{-1} = \left(\frac{\boldsymbol{\alpha_1}}{z_1} + \frac{\boldsymbol{\alpha_2}}{z_2} \right)^{-1},$$

Without loss of generality assume that $\boldsymbol{\alpha_1} \leq 1/2$. Using the notation of Section 5.2.1 we have that

$$\underline{H}_{\boldsymbol{\alpha_1}} \leq H_\alpha(x) \leq \overline{H}_{\boldsymbol{\alpha_1}}.$$

Further we know that $\overline{H}_{\boldsymbol{\alpha_1}}$ is decreasing and $\underline{H}_{\boldsymbol{\alpha_1}}$ is increasing in $\boldsymbol{\alpha_1}$. This leads to the following theorem.

Theorem 5.2.7 *Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n_{>0}$ with $\sum_{i=1}^n \alpha_i = 1$ and $x = (x_1, \ldots, x_n) \in \mathbb{R}^n_{>0}$ such that $A(x, \alpha) = A$ and $G(x, \alpha) = G < A$. Let futher $\underline{\alpha} = \min_{i=1,\ldots,n} \alpha_i$ and $y_1 \in [0, 1]$, $y_1' \in [1, \underline{\alpha}^{-1}]$ be the two solutions of*

$$Ay^{\underline{\alpha}} \left(\frac{1 - \underline{\alpha} y}{1 - \underline{\alpha}} \right)^{1-\underline{\alpha}} = G.$$

Then

$$A \frac{y_1(1 - \underline{\alpha} y_1)}{y_1(1 - 2\underline{\alpha}) + \underline{\alpha}} \leq H(x, \alpha) \leq A \frac{y_1'(1 - \underline{\alpha} y_1')}{y_1'(1 - 2\underline{\alpha}) + \underline{\alpha}}.$$

The first inequality reaches equality if and only if $x_l = y_1$ and $x_i = x_j \neq y_1$ for $i, j \neq l$, $1 \leq i \leq j \leq n$ for some l such that $\alpha_l = \underline{\alpha}$. The second inequality reaches equality if and only if $x_l = y_1'$ and $x_i = x_j \neq y_1'$ for $i, j \neq l$, $1 \leq i \leq j \leq n$ for some l such that $\alpha_l = \underline{\alpha}$.

Bounds on the geometric mean

As in the last section for $x = (x_1, \ldots, x_n) \in \mathbb{R}^n_{>0}$ with given arithmetic mean $A_\alpha(x) = 1$ and harmonic mean $H_\alpha(x) = G < 1$ we will bound the corresponding harmonic mean:

$$G_\alpha(x) = G(x, \alpha) = \prod_{i=1}^n x_i^{\alpha_i}.$$

We do this by upper respectively lower bounding the solutions to the following optimization problem.

$$\begin{aligned}
\text{minimize/maximize} \qquad & G_\alpha(x) = \prod_{i=1}^n x_i^{\alpha_i}, \\
\text{under the contraints} \qquad & A_\alpha(x) = \sum_{i=1}^n \alpha_i x_i = 1, \\
& H_\alpha(x) = \left(\sum_{i=1}^n \frac{\alpha_i}{x_i} \right)^{-1} = H, \\
& x \in X = \mathbb{R}^n_{>0} \subset \mathbb{R}^n.
\end{aligned}$$

The set $M := \{x \in X : H_\alpha(x) = H \wedge A_\alpha(x) = 1\}$ is again compact in X as the intersection of a closed and compact set. Using the same line of argumentation as in the last section we have that if $x \in M$ is an extremal point of $G_\alpha(x) = \prod_{i=1}^n x_i^{\alpha_i}$ then there exist parameters κ_1 and κ_2 such that

$$\frac{\partial}{\partial x_i} \left(\log(G(x, \alpha)) - \kappa_1 A(x, \alpha) - \kappa_2 H(x, \alpha)^{-1} \right) = 0, \quad 1 \leq i \leq n.$$

Computing the derivatives, we get the conditions

$$\frac{1}{x_i} - \kappa_1 - \kappa_2 \frac{1}{x_i^2} = 0, \quad 1 \leq i \leq n.$$

Again this is a polynomial of degree 2, and as such it has at most two roots z_1 and z_2. So for all $1 \leq i \leq n$, $x_i = z_1$ or $x_i = z_2$. As in our setting the x_i's can not all be equal, the two roots are not equal, i.e. $z_1 \neq z_2$. For an extremal point $x \in M$ let again $Z_1(x) = \{i : x_i = z_1\}$ and $Z_2(x) = \{i : x_i = z_2\}$ and $\boldsymbol{\alpha_1} := \sum_{i \in Z_1} \alpha_i$, $\boldsymbol{\alpha_2} := \sum_{i \in Z_2} \alpha_i = 1 - \boldsymbol{\alpha}_1$. Clearly

$$A_\alpha(x) = \sum_{i=1}^{n} \alpha_i x_i = \boldsymbol{\alpha_1} z_1 + \boldsymbol{\alpha_2} z_2 = 1,$$
$$H_\alpha(x) = \left(\sum_{i=1}^{n} \frac{\alpha_i}{z_i} \right)^{-1} = \left(\frac{\boldsymbol{\alpha_1}}{z_1} + \frac{\boldsymbol{\alpha_2}}{z_2} \right)^{-1} = H,$$

and

$$G_\alpha(x) = \prod_{i=1}^{n} x_i^{\alpha_i} = z_1^{\boldsymbol{\alpha_1}} z_2^{\boldsymbol{\alpha_2}}.$$

Without loss of generality assume that $\boldsymbol{\alpha_1} \leq 1/2$, then using the notation of Section 5.2.1

$$\underline{G}_{\boldsymbol{\alpha_1}} \leq G_\alpha(x) \leq \overline{G}_{\boldsymbol{\alpha_1}}.$$

As $\underline{G}_{\boldsymbol{\alpha}_1}$ is increasing and $\overline{G}_{\boldsymbol{\alpha}_1}$ decreasing in $\boldsymbol{\alpha_1}$ the following theorem follows:

Theorem 5.2.8 *Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n_{>0}$ with $\sum_{i=1}^n \alpha_i = 1$ and $x = (x_1, \ldots, x_n) \in \mathbb{R}^n_{>0}$ such that $A(x, \alpha) = A$ and $H(x, \alpha) = H < A$. Further let $\underline{\alpha} = \min \alpha_i$ and $y_1 \in [0, 1]$, $y_1' \in [1, \underline{\alpha}^{-1}]$ be the two solutions of*

$$A \frac{y(1 - \underline{\alpha} y)}{y(1 - 2\underline{\alpha}) + \underline{\alpha}} = H. \tag{5.2.29}$$

Then

$$A y_1'^{\underline{\alpha}} \left(\frac{1 - \underline{\alpha} y_1'}{1 - \underline{\alpha}} \right)^{1 - \underline{\alpha}} \leq G(x, \alpha) \leq A y_1^{\underline{\alpha}} \left(\frac{1 - \underline{\alpha} y_1}{1 - \underline{\alpha}} \right)^{1 - \underline{\alpha}}. \tag{5.2.30}$$

The first inequality reaches equality if and only if we have $x_l = y_1'$ and $x_i = x_j \neq y_1'$ for $i, j \neq l$, $1 \leq i \leq j \leq n$ for some l such that $\alpha_l = \underline{\alpha}$. The second inequality reaches equality if and only if $x_l = y_1$ and $x_i = x_j \neq y_1$ for $i, j \neq l$, $1 \leq i \leq j \leq n$ for some l such that $\alpha_l = \underline{\alpha}$.

Note that we can explicitly solve the quadratic equation given in Equation (5.2.29). Plugging in the solutions into Inequality (5.2.30) we get the following corollary:

Corollary 5.2.9 *Under the conditions given in Theorem 5.2.8 and defining*

$$\Delta := (H(1 - 2\underline{\alpha}) - A)^2 - 4\underline{\alpha}^2 HA.$$

we have that

$$\begin{aligned} G(x,\alpha) &\geq \left(\frac{A-H(1-2\underline{\alpha})+\sqrt{\Delta}}{2\alpha}\right)^{\underline{\alpha}} \left(\frac{A+H(1-2\underline{\alpha})-\sqrt{\Delta}}{2(1-\underline{\alpha})}\right)^{1-\underline{\alpha}}, \\ G(x,\alpha) &\leq \left(\frac{A-H(1-2\underline{\alpha})-\sqrt{\Delta}}{2\alpha}\right)^{\underline{\alpha}} \left(\frac{A+H(1-2\underline{\alpha})+\sqrt{\Delta}}{2(1-\underline{\alpha})}\right)^{1-\underline{\alpha}}. \end{aligned}$$

Using the notation from the corollary we can write

$$\Delta = (A-H)\big(A-H(1-2\underline{\alpha})^2\big). \tag{5.2.31}$$

As $(1-2\underline{\alpha})^2 \leq 1$ we have $(A-H)^2 \leq \Delta \leq \big(A-H(1-2\underline{\alpha})^2\big)^2$. We can prove the follwoing corollary:

Corollary 5.2.10 *Using the notation as above, if $H=1$ and $\underline{\alpha}=1/n$ we get*

$$G(x,\alpha) \leq e^{1/n}\big(A+1/n\big)^{(n-1)/n}.$$

Proof: Note that by (5.2.31) we have

$$\frac{A-H(1-2\underline{\alpha})-\sqrt{\Delta}}{2\underline{\alpha}} \leq \frac{A-(1-2\underline{\alpha})-(A-1)}{2\underline{\alpha}} = 1,$$

and

$$\begin{aligned} \frac{A+H(1-2\underline{\alpha})+\sqrt{\Delta}}{2(1-\underline{\alpha})} &= \frac{2A+(1-2\underline{\alpha})-(1-2\underline{\alpha})^2}{2(1-\underline{\alpha})} \\ &= (1-\underline{\alpha})^{-1}\big(A+\underline{\alpha}(1-2\underline{\alpha})\big) \\ &\leq (1-1/n)^{-1}\big(A+1/n\big). \end{aligned}$$

As $\left(\frac{1}{1-1/n}\right)^{n-1} = \left(1+\frac{1}{n-1}\right)^{n-1} < e$, we get that

$$G(x,\alpha) \leq e^{1/n}\big(A+1/n\big)^{(n-1)/n}.$$

□

Note that Lemma 5.0.4 is a direct consequence of the lemma above.

Chapter 6

Generating bases from random vectors

The initial motivation for this chapter was the development of a genetic algorithm to improve existing lattice reduction algorithms. Having a population of fairly reduced lattice bases, we were looking for ways to combine these bases to generate a new population of further reduced basis. For lattices of rank n the idea was to take a subset of $k_1 < n$ basis vectors from one basis and a subset of $k_2 < n - k_1$ basis vectors from a second basis, and complete them into a new basis containing the chosen basis vectors. The question was whether we can expect this kind of recombination to be successful with reasonable probability. The result of this chapter gives a partial answer to the question. Under the assumption that the coefficients of the chosen basis vectors with respect to some fixed basis are uniformly at random from a box $[-\beta, \beta]^n \cap \mathbb{Z}^n$, we compute the probability that it is possible to recombine two bases in this way for $\beta \to \infty$. Unfortunately the genetic algorithm as mentioned above did not prove to give a considerable improvement of existing lattice basis reduction algorithms.

We will see that the lattice vectors can be completed into a basis if and only if the corresponding rectangular matrix containing the coefficients of the vectors with respect to some fixed basis is *unimodular*:

Definition 6.0.11 *Let $1 \leq k \leq n$. An $n \times k$ integer matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$ is called* unimodular *if and only if the following three equivalent properties (see Lemma 6.4.2) are satisfied:*

1. *The matrix A can be completed into a unimodular $n \times n$ matrix, i.e. there exist $\mathbf{a}_{k+1}, \ldots, \mathbf{a}_n \in \mathbb{Z}^n$ such that $[\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathrm{GL}_n(\mathbb{Z})$.*

2. *There exists $\tilde{A} \in \mathbb{Z}^{k \times n}$ such that $\tilde{A}A$ is the $k \times k$ identity matrix $I_{k \times k}$, i.e. $\tilde{A}A = I_{k \times k}$.*

3. *The $k \times k$-minors of A are coprime, i.e. they do not have a common factor.*

For $1 \leq n \leq l$, an $n \times l$ integer matrix A is called unimodular, if and only if A^T is unimodular. These equivalences are true in general for matrices over principal ideal domains. For completeness we give a proof hereof in Section 6.4.1.

Let $\mathcal{L} \subset \mathbb{R}^m$ be a a lattice of rank n with basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. Then:

- Let $\mathbf{T} \subset \mathcal{L}$ be a set of $k < n$ lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and $A \in \mathbb{Z}^{n \times k}$ be such that $BA = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$. Then there exist $\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n \in \mathcal{L}$ such that $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$ forms a basis of $\mathcal{L}$ if and only if A can be completed into an $n \times n$ unimodular matrix, i.e. A is unimodular.

- Let $\mathbf{T}' \subset \mathcal{L}$ be a set of $l > n$ lattice vectors $\mathbf{v}'_1, \ldots, \mathbf{v}'_l$ and $A' \in \mathbb{Z}^{n \times l}$ be such that $BA' = [\mathbf{v}'_1, \ldots, \mathbf{v}'_l]$. Then $\mathbf{v}'_1, \ldots, \mathbf{v}'_l$ generate $\mathcal{L}$ if and only if there exists $\tilde{A}' \in \mathbb{Z}^{n \times l}$ such that $\tilde{A}'A' = I_{n \times n}$, i.e. A' is unimodular.

Under certain assumptions on the coefficients of the respective matrices A we can compute the probability that $\mathbf{v}_1, \ldots, \mathbf{v}_k$ can be completed into a lattice basis. Under the same assumptions we can compute the probability that the vectors $\mathbf{v}'_1, \ldots, \mathbf{v}'_l$ generate $\mathcal{L}$ respectively. More concretely, for $1 \leq k < n$, we prove that the probability that a $k \times n$ integer matrix is unimodular when the coefficients are chosen uniformly at random from an interval $[-\beta, \beta] \cap \mathbb{Z}$ tends to $\prod_{j=n-k+1}^{n} \zeta(j)^{-1}$ as β goes to infinity, where $\zeta(j)$ denotes the Riemann zeta function. This result emerged out of a close collaboration with G. Maze and J. Rosenthal and has been published in [MRW11].

The result generalizes a well known result due to Cesaro [Ces84] which states that the 'probability' that two randomly chosen integers are coprime equals $\zeta(2)^{-1} = \frac{6}{\pi^2}$. More generally it is known that the probability of n integers being coprime is $\zeta(n)^{-1}$ [Ces84, Leh00, Nym72]. Note that this corresponds to our cases when we choose $k = 1$. Our result does not build on the result by Cesaro and as such gives an independent proof also of the above facts.

The chapter is organized as follows. In Section 6.1 we quickly review the notion of natural density. Section 6.2 then contains the main result along with its proof. In Section 6.3 we give some further remarks.

6.1 Preliminaries

For $\beta \in \mathbb{R}$, we denote by $[\![-\beta, \beta]\!]$ the set of integers lying inside the interval $[-\beta, \beta]$, i.e.

$$[\![-\beta, \beta]\!] := [-\beta, \beta] \cap \mathbb{Z}.$$

Consequently $[\![-\beta, \beta]\!]^{k \times n}$ denotes the set of $k \times n$ integer matrices whose coefficients are elements of $[\![-\beta, \beta]\!]$. Let $\mathbf{S} \subset \mathbb{Z}^{k \times n}$. It is clear that the probability that an element chose uniformly at random from $[\![-\beta, \beta]\!]^{k \times n}$ is in $\mathbf{S}$ equals the cardinality $\mathbf{S} \cap [\![-\beta, \beta]\!]^{k \times n}$ divided by the cardinality of $[\![-\beta, \beta]\!]^{k \times n}$, i.e.

$$\frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]^{k \times n}|}{|[\![-\beta, \beta]\!]^{k \times n}|}.$$

If the limit for $\beta \to \infty$ hereof exists, it basically tells us how frequent the elements of $\mathbf{S}$ are in $\mathbb{Z}^{k\times n}$. This is exactly the concept of natural density.

Definition 6.1.1 *For $1 \leq k \leq n \in \mathbb{N}$ and a set $\mathbf{S} \subset \mathbb{Z}^{k\times n}$ we define the* upper natural density *as*

$$\overline{\mathbb{D}}(\mathbf{S}) := \limsup_{\beta\to\infty} \frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]^{k\times n}|}{|[\![-\beta, \beta]\!]^{k\times n}|},$$

the lower natural density *as*

$$\underline{\mathbb{D}}(\mathbf{S}) := \liminf_{\beta\to\infty} \frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]^{k\times n}|}{|[\![-\beta, \beta]\!]^{k\times n}|},$$

and in the case where they are the same the natural density

$$\mathbb{D}(\mathbf{S}) := \overline{\mathbb{D}}(\mathbf{S}) = \underline{\mathbb{D}}(\mathbf{S}).$$

It is important to note that this limit does not always exist. An often used example thereof is given in the following remark.

Remark 6.1.2 *Let $k = n = 1$ and consider the set of integers whose leading digit is one:*

$$\mathbf{S} := \left\{x \in \mathbb{Z} : |x| = 10^l + r \text{ with } l \in \mathbb{N}, r < 10^l\right\}.$$

Then

$$\lim_{j\to\infty} \frac{|\mathbf{S} \cap [\![-2\cdot 10^j, 2\cdot 10^j]\!]|}{|[\![-2\cdot 10^j, 2\cdot 10^j]\!]|} = \lim_{j\to\infty} \frac{2\sum_{i=0}^{j} 10^i}{4\cdot 10^j + 1} = \frac{10}{18}.$$

and

$$\lim_{j\to\infty} \frac{|\mathbf{S} \cap [\![-10^j + 1, 10^j - 1]\!]|}{|[\![-10^j + 1, 10^j - 1]\!]|} = \lim_{j\to\infty} \frac{2\sum_{i=0}^{j-1} 10^i}{2\cdot 10^j - 1} = \frac{1}{9}.$$

So we have two convergent subsequences with different limits. As a consequence

$$\limsup_{\beta\to\infty} \frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]|}{|[\![-\beta, \beta]\!]|} \neq \liminf_{\beta\to\infty} \frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]|}{|[\![-\beta, \beta]\!]|},$$

i.e. the natural density of the given set $\mathbf{S}$ is not defined.

In the following lemma we summarize two properties of the natural density that we will need at some point in the sequel. It is based [MRW11] on the fact that for real sequences $(a_i)_{i\in\mathbb{N}}$, $(b_i)_{i\in\mathbb{N}}$ it holds that

$$\begin{aligned} \liminf_{i\in\mathbb{N}} a_i + \liminf_{i\in\mathbb{N}} b_i &\leq \liminf_{i\in\mathbb{N}} (a_i + b_i), \\ \limsup_{i\in\mathbb{N}} (a_i + b_i) &\leq \limsup_{i\in\mathbb{N}} a_i + \limsup_{i\in\mathbb{N}} b_i. \end{aligned} \tag{6.1.1}$$

Lemma 6.1.3 *We use the notation from above.*

1. *Let* $\mathbf{S} = \mathbf{S}_1 \sqcup \mathbf{S}_2 \subset \mathbb{Z}^{k\times n}$ *be the disjoint union of two sets* $\mathbf{S}_1$ *and* $\mathbf{S}_2$ *and suppose that* $\mathbb{D}(\mathbf{S}_1)$ *exists. Then*

$$\mathbb{D}(\mathbf{S}_1) + \underline{\mathbb{D}}(\mathbf{S}_2) \leq \underline{\mathbb{D}}(\mathbf{S}) \leq \overline{\mathbb{D}}(\mathbf{S}) \leq \mathbb{D}(\mathbf{S}_1) + \overline{\mathbb{D}}(\mathbf{S}_2).$$

2. *Let* $\mathbf{S}_1$ *be as above, then* $\mathbb{D}\big(\mathbb{Z}^{k\times n}\backslash\mathbf{S}_1\big) = \mathbb{D}(\mathbf{S}_1^c) = 1 - \mathbb{D}(\mathbf{S}_1)$.

3. *Let* $(\mathbf{S}_i)_{i\in\mathbb{N}}$ *be a sequence of sets in* $\mathbb{Z}^{k\times n}$. *Then*

$$\overline{\mathbb{D}}\left(\bigcup_{i\in\mathbb{N}} \mathbf{S}_i\right) \leq \sum_{i\in\mathbb{N}} \overline{\mathbb{D}}(\mathbf{S}_i).$$

Note that for $\beta \in \mathbb{R}$ and $\mathbf{S} \subset \mathbb{Z}^{k\times n}$ we have that

$$\frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]^{k\times n}|}{|[\![-\beta, \beta]\!]^{k\times n}|} = \frac{|\mathbf{S} \cap [\![-\lfloor\beta\rfloor, \lfloor\beta\rfloor]\!]^{k\times n}|}{|[\![-\lfloor\beta\rfloor, \lfloor\beta\rfloor]\!]^{k\times n}|}.$$

Thus in order to investigate the limit behaviour of the above it is enough to consider $\beta \in \mathbb{Z}$.

6.2 Natural density of rectangular unimodular matrices

We will now compute the natural density of the following set $\mathbf{S} \subset \mathbb{Z}^{k\times n}$:

$$\mathbf{S} := \left\{A \in \mathbb{Z}^{k\times n} : A \text{ is unimodular}\right\}.$$

Let us first fix some notation. By $\mathcal{P} \subset \mathbb{Z}$ we denote the set of all prime numbers. Given a set $\Omega \subset \mathcal{P}$ we use the following notation:

$$N_\Omega := \prod_{p\in\Omega} p, \tag{6.2.2}$$

$$\mathbf{S}_\Omega := \left\{A \in \mathbb{Z}^{k\times n} : A \text{ has full rank modulo } p,\ p \in \Omega\right\}, \tag{6.2.3}$$

$$\mathbf{F}_\Omega := \left\{A \in \mathbb{Z}_{N_\Omega}^{k\times n} : A \text{ has full rank modulo } p,\ p \in \Omega\right\}, \tag{6.2.4}$$

where $\mathbb{Z}_{N_\Omega}$ is the residue class ring $\mathbb{Z}/N_\Omega\mathbb{Z}$. In the case where Ω is just the set of the first t primes $p_1 < \cdots < p_t$ we will write $\mathbf{S}_t$, $\mathbf{F}_t$ respectively for the sets defined in (6.2.3), (6.2.4) and N_t for the product as defined in (6.2.2). Further let

$$\varphi_\Omega : \mathbb{Z}^{k\times n} \longrightarrow \mathbb{Z}_{N_\Omega}^{k\times n}, \tag{6.2.5}$$

be the (coefficientwise) natural quotient map.

Let us give the strategy to compute the natural density of $\mathbf{S}$. For a finite set $\Omega \in \mathcal{P}$ we compute $|\mathbf{F}_\Omega|$ (Lemma 6.2.1). Note that the elements in $\mathbf{S}_\Omega$ are exactly the integer matrices for which the gcd of the $k \times k$ minors is coprime to N_Ω. Hence, $\mathbf{S}_\Omega$ is exactly

the preimage of $\mathbf{F}_\Omega$ under φ_Ω, i.e. $\mathbf{S}_\Omega = \varphi^{-1}(\mathbf{F}_\Omega)$. By upper and lower bounding $|\varphi^{-1}(\mathbf{F}_\Omega)| \cap [\![-\beta, \beta]\!]$ we can compute the natural density of $\mathbf{S}_\Omega$ and hence $\mathbf{S}_t$ (Lemma 6.2.3, Corollary 6.2.4). While at first sight it is tempting to conclude directly that $\mathbb{D}(\mathbf{S}) = \mathbb{D}(\mathbf{S}_\mathcal{P}) = \lim_{t\to\infty} \mathbb{D}(\mathbf{S}_t)$, it needs some work to show that this is in fact the case (Proposition 6.2.5).

We start with the following lemma.

Lemma 6.2.1 *Let Ω be a finite set of prime numbers and N_Ω the product of these as above. Then the following holds*

$$|\mathbf{F}_\Omega| = N_\Omega^{kn} \prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right).$$

Proof: By the Chinese remainder theorem we have an isomorphism

$$\mathbb{Z}_{N_\Omega}^{k\times n} \cong \prod_{p\in\Omega} \mathbb{Z}_p^{k\times n}. \tag{6.2.6}$$

Now, for $A \in \mathbb{Z}_{N_\Omega}^{k\times n}$ we have that $A \in \mathbf{F}_\Omega$ if and only if for all $p \in \Omega$ it has full rank mod p, i.e. $(A \mod p) \in \mathbf{F}_{\{p\}} = \mathbb{Z}_p^{k\times n}$. Hence $\mathbf{F}_\Omega \subset \mathbb{Z}_{N_\Omega}^{k\times n}$ is the preimage of $\prod_{p\in\Omega} \mathbf{F}_{\{p\}} \subset \prod_{p\in\Omega} \mathbb{Z}_p^{k\times n}$ under the isomorphism given in (6.2.6). Hence, $\|\mathbf{F}_\Omega\| = \prod_{p\in\Omega} |\mathbb{Z}_p^{k\times n}|$. The cardinality of $\mathbf{F}_{\{p\}}^{k\times n}$ is not hard to compute. It is well known that for p prime,

$$\left|\mathbf{F}_{\{p\}}\right| = \prod_{i=0}^{k-1} \left(p^n - p^i\right) = p^n \prod_{i=0}^{k-1} \left(1 - \frac{1}{p^{n-i}}\right) = p^n \prod_{i=n-k+1}^{n} \left(1 - \frac{1}{p^i}\right).$$

Consequently

$$|\mathbf{F}_\Omega| = \prod_{p\in\Omega} |\mathbf{F}_{\{p\}}| = N_\Omega^{kn} \prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right).$$

□

The next corollary follows immediately.

Corollary 6.2.2 *Let N_t be the product of the first t primes, $N_t = p_1 \cdots p_t$ and $\mathbf{F}_t \subset \mathbb{Z}_{N_t}^{k\times n}$ as above. Then*

$$|\mathbf{F}_t| = N_t^{kn} \prod_{i=n-k+1}^{n} \prod_{j=1}^{t} \left(1 - \frac{1}{p_j^i}\right).$$

We can now compute the natural density of $\mathbf{S}_\Omega$, which is the preimage of $\mathbf{F}_\Omega$ under φ_Ω as in (6.2.5).

Lemma 6.2.3 *Using the notation from above we have*

$$\mathbb{D}(\mathbf{S}_\Omega) = \prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right).$$

Proof: For $\beta \in \mathbb{N}$ let

$$D_\Omega(\beta) := \frac{|\mathbf{S}_\Omega \cap [\![-\beta,\beta]\!]^{k\times n}|}{|[\![-\beta,\beta]\!]^{k\times n}|} = \frac{|\mathbf{S}_\Omega \cap [\![-\beta,\beta]\!]^{k\times n}|}{(2\beta+1)^{kn}}.$$

The goal is to compute $\lim_{\beta\to\infty} D_\Omega(\beta)$. Write $\beta = \alpha N_\Omega + r$ with $\alpha \in \mathbb{N}$ and $0 \le r < N_\Omega$. Then we can write $[\![-\beta,\beta]\!]^{k\times n}$ as the disjunct union

$$[\![-\beta,\beta]\!]^{k\times n} = [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n} \sqcup \left([\![-\beta,\beta]\!]^{k\times n} \backslash [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}\right).$$

As a consequence,

$$|\mathbf{S}_\Omega \cap [\![-\beta,\beta]\!]^{k\times n}| = |\mathbf{S}_\Omega \cap [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}| + |\mathbf{S}_\Omega \cap \left([\![-\beta,\beta]\!]^{k\times n} \backslash [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}\right)|.$$

With $|[\![-\beta,\beta]\!]^{k\times n} \backslash [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}| = (2r)^{kn} < (2N_\Omega)^{kn}$,

$$0 \le \frac{|\mathbf{S}_\Omega \cap \left([\![-\beta,\beta]\!]^{k\times n} \backslash [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}\right)|}{(2\beta+1)^{kn}} < \underbrace{\frac{2N_\Omega^{kn}}{(2\beta+1)^{kn}}}_{=:\rho(\beta)}.$$

So we can write,

$$D_\Omega(\beta) \le \frac{|\mathbf{S}_\Omega \cap [\![-\beta,\beta]\!]^{k\times n}|}{(2\beta+1)^{kn}} = \frac{|\mathbf{S}_\Omega \cap [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}|}{(2\beta+1)^{kn}} + \rho(\beta). \tag{6.2.7}$$

Consider the map taking the quotient modulo N_Ω restricted to $[\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}$:

$$\tilde{\varphi} = \varphi|_{[\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}} : [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n} \to \mathbb{Z}_{N_\Omega}^{k\times n}.$$

It is clear that $A \in \mathbf{S}_\Omega \cap [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}$ if it is in the preimage $\tilde{\varphi}^{-1}(\mathbf{F}_{N_\Omega})$ of $\mathbf{F}_{N_\Omega}$. The number of preimages of an element $T \in \mathbf{F}_{N_\Omega}$ is bounded by

$$(2\alpha)^{kn} \le |\tilde{\varphi}^{-1}(T)| \le (2\alpha+1)^{kn},$$

depending on how many of its coefficients are zero modulo N_Ω. Hence,

$$\frac{(2\alpha)^{kn}|\mathbf{F}_{N_\Omega}|}{(2\beta+1)^{kn}} \le \frac{|\mathbf{S}_\Omega \cap [\![-\alpha N_\Omega, \alpha N_\Omega]\!]^{k\times n}|}{(2\beta+1)^{kn}} \le \frac{(2\alpha+1)^{kn}|\mathbf{F}_{N_\Omega}|}{(2\beta+1)^{kn}}. \tag{6.2.8}$$

Combining (6.2.7), (6.2.8) and Lemma 6.2.1 we get

$$D_\Omega(\beta) \le \frac{(2\alpha+1)^{kn} N_\Omega^{kn}}{(2\beta+1)^{kn}} \prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right) + \rho(\beta),$$

$$D_\Omega(\beta) \ge \frac{(2\alpha)^{kn} N_\Omega^{kn}}{(2\beta+1)^{kn}} \prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right).$$

Note that

$$(2\beta - N_\Omega)^{kn} < (2\alpha)^{kn} N_\Omega^{kn} < 2\beta^{kn} < (2\alpha+1)^{kn} N_\Omega^{kn} < (2\beta + N_\Omega)^{kn}.$$

So with $\rho(\beta) \overset{\beta\to\infty}{\longrightarrow} 0$ we get that

$$\prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right) \leq \liminf_{\beta\to\infty} D_\Omega(\beta) \leq \limsup_{\beta\to\infty} D_\Omega(\beta) \leq \prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right).$$

and hence

$$\underline{\mathbb{D}}(\mathbf{S}_\Omega) = \overline{\mathbb{D}}(\mathbf{S}_\Omega) = \mathbb{D}(\mathbf{S}_\Omega) = \prod_{i=n-k+1}^{n} \prod_{p\in\Omega} \left(1 - \frac{1}{p^i}\right).$$

□

Again, the following corollary gives the situation when the set of primes contains just the first $t > 1$ primes.

Corollary 6.2.4 *Using the notation from above we have*

$$\mathbb{D}(\mathbf{S}_t) = \prod_{i=n-k+1}^{n} \prod_{j=1}^{t} \left(1 - \frac{1}{p_j^i}\right).$$

We will use this corollary to compute the natural density of $\mathbf{S}$. Note that $\mathbf{S} = \mathbf{S}_\mathcal{P} = \lim_{t\to\infty} \mathbf{S}_t$.

Proposition 6.2.5 *Let $1 \leq k < n$, then the natural density $\mathbb{D}(\mathbf{S})$ of the set $\mathbf{S} = \{A \in \mathbb{Z}^{k\times n} : A \text{ is unimodular}\}$ equals*

$$\mathbb{D}(\mathbf{S}) = \prod_{i=n-k+1}^{n} \prod_{p\in\mathcal{P}} \left(1 - \frac{1}{p^i}\right).$$

Proof: Let again $\beta \in \mathbb{N}$. We have to determine the limit behaviour of the following,

$$D(\beta) := \frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]^{k\times n}|}{|[\![-\beta, \beta]\!]^{k\times n}|} = \frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]^{k\times n}|}{(2\beta+1)^{kn}},$$

when β goes to infinity. Note that $\forall t \in \mathbb{N} : \mathbf{S} \subset \mathbf{S}_t$ and hence $\mathbf{S} = \mathbf{S}_t \backslash (\mathbf{S}_t \backslash \mathbf{S})$. So for $t \in \mathbb{N}$ we get

$$D(\beta) = \frac{|\mathbf{S}_t \cap [\![-\beta, \beta]\!]^{k\times n}|}{(2\beta+1)^{kn}} - \frac{|(\mathbf{S}_t \backslash \mathbf{S}) \cap [\![-\beta, \beta]\!]^{k\times n}|}{(2\beta+1)^{kn}}.$$

By Corollary 6.2.4 and Lemma 6.1.3 we have

$$\mathbb{D}(\mathbf{S}_t) + \underline{\mathbb{D}}(\mathbf{S}_t \backslash \mathbf{S}) \leq \liminf_{\beta\to\infty} D(\beta) \leq \limsup_{\beta\to\infty} D(\beta) \leq \mathbb{D}(\mathbf{S}_t) + \overline{\mathbb{D}}(\mathbf{S}_t \backslash \mathbf{S}),$$

for all $t \in \mathbb{N}$. In particular,

$$\mathbb{D}(\mathbf{S}_t) \leq \liminf_{\beta\to\infty} D(\beta) \leq \limsup_{\beta\to\infty} D(\beta) \leq \mathbb{D}(\mathbf{S}_t) + \overline{\mathbb{D}}(\mathbf{S}_t\backslash\mathbf{S}).$$

Clearly $\lim_{t\to\infty} \mathbb{D}(\mathbf{S}_t)$ exists being the Euler product of the inverse of the Riemann zeta function and consequently

$$\lim_{t\to\infty} \mathbb{D}(\mathbf{S}_t) \leq \liminf_{\beta\to\infty} D(\beta),$$

and by (6.1.1)

$$\limsup_{\beta\to\infty} D(\beta) \leq \limsup_{t\to\infty} \left(\mathbb{D}(\mathbf{S}_t) + \overline{\mathbb{D}}(\mathbf{S}_t\backslash\mathbf{S})\right) \leq \lim_{t\to\infty} \mathbb{D}(\mathbf{S}_t) + \limsup_{t\to\infty} \overline{\mathbb{D}}(\mathbf{S}_t\backslash\mathbf{S}).$$

If we can show that $\limsup_{t\to\infty} \overline{\mathbb{D}}(\mathbf{S}_t\backslash\mathbf{S}) = 0$, we can conclude that $\mathbb{D}(\mathbf{S}) = \lim_{\beta\to\infty} D(\beta) = \lim_{t\to\infty} \mathbb{D}(\mathbf{S}_t)$ which gives the result.

Note that $(\mathbf{S}_t\backslash\mathbf{S}) \subset \bigcup_{p>p_t} \left(\mathbb{Z}^{k\times n}\backslash\mathbf{S}_{\{p\}}\right)$. By Lemma 6.1.3,

$$\begin{aligned}
\overline{\mathbb{D}}(\mathbf{S}_t\backslash\mathbf{S}) &\leq \overline{\mathbb{D}}\left(\bigcup_{p>p_t} \mathbb{Z}^{k\times n}\backslash\mathbf{S}_{\{p\}}\right) \\
&\leq \sum_{p>p_t} \overline{\mathbb{D}}\left(\mathbb{Z}^{k\times n}\backslash\mathbf{S}_{\{p\}}\right) = \sum_{p>p_t} \mathbb{D}\left(\mathbf{S}^c_{\{p\}}\right) \\
&= \sum_{p>p_t} \left(1 - \mathbb{D}\left(\mathbf{S}_{\{p\}}\right)\right) = \sum_{p>p_t} \left(1 - \prod_{i=n-k+1}^{n} \left(1 - \frac{1}{p^i}\right)\right).
\end{aligned}$$

It can be shown that for reals $0 < x_i < 1$, it holds that $1 - \prod(1 - x_i) \leq \sum x_i$. Using the formulas for geometric series we get

$$1 - \prod_{i=n-k+1}^{n} \left(1 - \frac{1}{p^i}\right) \leq \sum_{i=n-k+1}^{n} \frac{1}{p^i} < \frac{1}{p^{n-k}(p-1)} < \frac{2}{p^2},$$

and hence,

$$\overline{\mathbb{D}}(\mathbf{S}_t\backslash\mathbf{S}) < \sum_{p>p_t} \frac{2}{p^2}.$$

As tail of a convergent sum this converges to zero and in particular $\limsup_{t\to\infty} \overline{\mathbb{D}}(\mathbf{S}_t\backslash\mathbf{S}) = 0$. We conclude that

$$\mathbb{D}(\mathbf{S}) = \lim_{t\to\infty} \mathbb{D}(\mathbf{S}_t) = \prod_{i=n-k+1}^{n} \prod_{p\in\mathcal{P}} \left(1 - \frac{1}{p^i}\right),$$

and the claim follows . □

For $s \in \mathbb{C}$ with real part greater than one, $\Re(s) > 1$, the well known Riemann zeta function $\zeta(s)$ is given by the convergent series

$$\zeta(s) = \sum_{i=1}^{\infty} i^{-s}.$$

Euler's formula gives the connection to the set of prime numbers:

$$\zeta(s) = \prod_{p \in \mathcal{P}} \left(1 - \frac{1}{p^s}\right)^{-1}. \tag{6.2.9}$$

So Proposition 6.2.5 gives the main result:

Theorem 6.2.6 *Let $1 \leq k < n$, then the natural density $\mathbb{D}(\mathbf{S})$ of the set $\mathbf{S} = \{A \in \mathbb{Z}^{k\times n} : A \text{ is unimodular}\}$ equals*

$$\mathcal{D}_{k,n} := \mathbb{D}(\mathbf{S}) = \prod_{i=n-k+1}^{n} \frac{1}{\zeta(i)}.$$

In the case when $k = n$,

$$\mathcal{D}_{n,n} := \mathbb{D}(\mathbf{S}) = 0.$$

Proof: The first part readily follows from Proposition 6.2.5 and (6.2.9). In the case of $k = n$ the following argument leads to the result. Let again $\beta \in \mathbb{N}$. By the Lagrange expansion of the determinant, each $n \times n$ matrix with $n^2 - 1$ entries in $[\![-\beta, \beta]\!]$ can be extended to a unimodular matrix by at most two values, as the determinant must be ± 1. Consequently,

$$|\mathbf{S} \cap [\![-\beta, \beta]\!]^{n\times n}| \leq 2(2\beta + 1)^{n^2-1},$$

and hence

$$\limsup_{\beta\to\infty} \frac{|\mathbf{S} \cap [\![-\beta, \beta]\!]^{n\times n}|}{|[\![-\beta, \beta]\!]|^{n\times n}} \leq \limsup_{\beta\to\infty} \frac{2(2\beta + 1)^{n^2-1}}{(2\beta + 1)^{n^2}} = \limsup_{\beta\to\infty} \frac{2}{2\beta + 1} = 0.$$

□

6.3 Conclusion and extensions

Coming back to the initially mentioned connection to lattice theory, let again $\mathcal{L} \in \mathbb{R}^m$ be a lattice with basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. Let $l \geq n$ and $\mathbf{v}_1, \ldots, \mathbf{v}_l$ random lattice vectors in the sense that the coefficients of $\mathbf{v}_i$ with respect to the basis B are uniformly at random from $[\![-\beta, \beta]\!]$. Then it is immediate that for large β, the probability that $\mathbf{v}_1, \ldots, \mathbf{v}_l$ generate $\mathcal{L}$ tends to $\mathcal{D}_{n,l}$.

However this observation is of limited interest, as we do no know how to sample vectors $\mathbf{v}_1, \ldots, \mathbf{v}_l$ randomly in the above mentioned sense other than idependently uniformly

at random sampling the coefficients from $[\![-\beta, \beta]\!]$ and computing the corresponding lattice points. More interesting is the situation when the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_l$ are sampled uniformly at random from $\mathcal{L} \cap [-\beta, \beta]^n$. Let ψ is the isomorphism $\psi : \mathbb{R}^n \to \mathbb{R}^n$ defined by $\psi(\mathbf{e}_i) = \mathbf{b}_i$, $i = 1, \ldots, n$, where $\mathbf{e}_i$ denotes the i-th unit vector in $\mathbb{R}^n$. Then picking random elements in $\mathcal{L} \cap [-\beta, \beta]^n$ corresponds to picking random elements inside $X := \psi^{-1}([-\beta, \beta]^n) \cap \mathbb{Z}^n$. In the case where the basis B is reduced, i.e. its basis vectors are not too far from orthogonal, X is a not too skewed parallelepiped. Fontein and Wocjan [FW12, Conjecture 3.1] conjecture that if β is chosen large enough (dependent on the volume of the lattice and the length of the corresponding shortest vector), the probability that $n+1$ vectors chosen uniformly at random from $\mathcal{L} \cap [-\beta, \beta]^n$ generate $\mathcal{L}$ is not much less than $\mathcal{D}_{n,n+1} = \prod_{i=2}^{n+1} \frac{1}{\zeta(i)}$, being somehow what we would expect from the considerations above.

Further we would like to mention the work by Maze [Maz11]. It is not hard to see that a matrix $A \in \mathbb{Z}^{k \times n}$ is unimodular if and only if its Hermite normal form (HNF) is of the following form:

$$\begin{pmatrix} 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \\ \vdots & & \vdots & 0 & 1 & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & 0 & \ldots & 0 & 1 \end{pmatrix}. \tag{6.3.10}$$

While our result directly gives the natural density of integer matrices with HNF as in (6.3.10), he more generally computes the natural density distribution of the HNF with given diagonal elements.

Let us also mention an extension of our work by Guo and Yang [GY13]. By Remark 6.4.3, the equivalences given in Lemma 6.4.2 defining unimodular matrices over a ring R is not restricted to PIDs. In particular, they also hold for the polynomial ring $R = \mathbb{F}[x]$ over a field $\mathbb{F}$. While it is not clear how to extend the concept of natural density to the ring $R = \mathbb{F}[x]$ in general, Guo and Yang [GY13] show how this can be done in the case where $\mathbb{F}$ is a finite field with q elements, i.e. $\mathbb{F} = \mathbb{F}_q$. They show that the 'probability' that a random $k \times n$ matrix in $\mathbb{F}_q[x]$ is unimodular equals

$$\prod_{i=0}^{k-1} \left(1 - \frac{1}{q^{n-k+i}}\right).$$

Only after the publication of our results in [MRW11] we discovered that the main result has already been proven by S. Elizalde and K. Woods [EW07]. They examine the probability that at set of k vectors in $\mathbb{Z}^n$ form a primitive set, when the coefficients of the vectors are chosen uniformly at random from intervals of length β, where the intervals are not too far from the origin. They show that this probability tends to $\prod_{i=n-k+1}^{n} \zeta(i)^{-1}$ as β goes to infinity. A set of k vectors in $\mathbb{Z}^n$ being primitive is thereby equivalent to the corresponding $k \times n$ matrix being unimodular.

6.4 Appendix

6.4.1 Rectangular unimodular matrices over a PID

We prove the equivalences of Definition 6.0.11. The equivalences hold for matrices over principal ideal domains in general and can be shown using the existence and properties of the Smith Normal Form.

Theorem 6.4.1 ([HH83]) *Let A be a $k \times n$ matrix with $k \leq n$ over some principal ideal domain R. Then there exist invertible matrices $X \in \mathrm{GL}_k(\mathbb{Z})$ and $Y \in \mathrm{GL}_n(\mathbb{Z})$ such that*

$$XAY = \begin{pmatrix} d_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & d_2 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & \dots & 0 & d_k & 0 & \dots & 0 \end{pmatrix} \in R^{k \times n}, \tag{6.4.11}$$

where $d_1 | d_2 | \dots | d_k$.

Transposing the matrices in (6.4.11) shows the corresponding result in the case where $n > k$. The form in (6.4.11) is unique up to multiplication of the d_i's with units in R and is called the *Smith Normal Form* (SNF) of a matrix. The d_j's are called *elementary divisors* of A and it holds that $\prod_{j=1}^{i} d_j$ equals the greatest common divisor of all $j \times j$ minors of A (e.g. [vdW03]).

Lemma 6.4.2 *Let $1 \leq k \leq n$ and $A = [\mathbf{a}_1, \dots, \mathbf{a}_k]$ be an $n \times k$ matrix over some principal ideal domain R. Then the following three properties are equivalent:*

1. *A can be completed into a unimodular $n \times n$ matrix, i.e. there exist $\mathbf{a}_{k+1}, \dots, \mathbf{a}_n \in R^n$ such that $[\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathrm{GL}_n(R)$.*
2. *There exists $B \in R^{k \times n}$ such that $B \cdot A$ is the $k \times k$ identity matrix $I_{k \times k}$, i.e. $A \cdot B = I_{k \times k}$.*
3. *The $k \times k$-minors of A are coprime, i.e. they do not have a common factor.*

Proof: 1. $\Rightarrow$ 2.: Let $[\mathbf{b}_1, \dots, \mathbf{b}_n]^T$ be the inverse of $[\mathbf{a}_1, \dots, \mathbf{a}_n]$. Then $B = [\mathbf{b}_1, \dots, \mathbf{b}_k]^T$ satisfies $BA = I_{k \times k}$.
2. $\Rightarrow$ 3.: This follows from Cauchy-Binet's theorem (e.g. [Bro89]) which implies that the determinant of AB equals the sum of the products of a $k \times k$ minor of A with one of B.
3. $\Rightarrow$ 2.: The $k \times k$ minors of A being coprime implies by Theorem 6.4.1 that its Smith normal Form is of the form $(I_{k \times k} | 0_{k \times n})^T$, i.e. that there exist unitary X and Y such that $XA^TY = (I_{k \times k} | 0_{k \times n})$. Let Y' be the truncation of Y to the first k columns. Then it follows that $A^TY' = X^{-1}$ and hence $A^TY'X = I_{k \times k}$.
3. $\Rightarrow$ 1.: Let $XA^TY = (I_{k \times k} | 0_{k \times n})$ again be the SNF of A^T and $B = Y'X$ as above. Let $\tilde{\mathbf{y}}_{k+1}, \dots, \tilde{\mathbf{y}}_n$ the last $n - k$ columns of $(Y^{-1})^T$. Then $[\mathbf{a}_1, \dots, \mathbf{a}_k, \tilde{\mathbf{y}}_{k+1}, \dots, \tilde{\mathbf{y}}_n]$ forms a unitary matrix with inverse given by B concatenated with the last $n-k$ columns of Y^T.
□

We will call a $k \times n$ matrix over a PID satisfying these equivalent properties *unimodular*. While it is relatively easy to see that these three properties are equivalent in the case of R being a PID, this does not hold for PID's exclusively:

Remark 6.4.3 *Let $\mathbb{F}$ be a field and R the commutative polynomial ring $\mathbb{F}[x_1, \ldots, x_l]$. Then the three properties from Lemma 6.4.2 can be shown to be equivalent as well [Qui76, Sus76, YP84].*

Bibliography

[AEVZ02] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM.

[Ajt98] M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 10–19, New York, NY, USA, 1998. ACM.

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, STOC '01, pages 601–610, New York, NY, USA, 2001. ACM.

[Bab86] L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.

[Blö00] J. Blömer. Closest vectors, successive minima, and dual HKZ-bases of lattices. In *Automata, languages and programming (Geneva, 2000)*, volume 1853 of *Lecture Notes in Computer Science*, pages 248–259, Berlin, 2000. Springer.

[Bro89] J. Broida. *A comprehensive introduction to linear algebra*. Addison-Wesley, Redwood City, Calif, 1989.

[Ces84] E. Cesàro. *Probabilité de certains faits arithmétiques*. Number Bd. 4. Gauthier-Villars, 1884.

[CN11] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *Advances in Cryptology – Proceedings of ASIACRYPT '11*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2011.

[CSB87] J.H. Conway, N.J. Sloane, and E. Bannai. *Sphere-packings, lattices, and groups.* Springer, New York, 1987.

[DKS98] I. Dinur, G. Kindler, and S. Safra. Approximating-CVP to within almost-polynomial factors is NP-hard. In *Proceedings of 39th Annual Symposium on Foundations of Computer Science, '98*, pages 99–109. IEEE, 1998.

[EW07] S. Elizalde and K. Woods. The probability of choosing primitive sets. *Journal of Number Theory*, 125(1):39 – 49, 2007.

[FP85] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation*, 44:463–463, 1985.

[FSW13] F. Fontein, M. Schneider, and U. Wagner. A polynomial time version of LLL with deep insertions. In *Preproceedings of the International Workshop on Coding and Cryptography, WCC '13*, 2013.

[FW12] F. Fontein and P. Wocjan. On the probability of generating a lattice. `arXiv:1211.6246`, 2012. Submitted.

[GG00] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000.

[GM03] D. Goldstein and A. Mayer. On the equidistribution of hecke points. *Forum Mathematicum*, 15(2):165–189, 2003.

[GN08] N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *Advances in Cryptology – Proceedings of EUROCRYPT '08*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 2008.

[GY13] X. Guo and G. Yang. The probability of rectangular unimodular matrices over $\mathrm{F}_q[x]$. *Linear Algebra and its Applications*, 438(6):2675 – 2682, 2013.

[HH83] B. Hartley and T. O. Hawkes. *Rings modules and linear algebra.* Chapman and Hall, London, New York, 1983.

[HPS11] G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *Advances in Cryptology - CRYPTO '11*, volume 6841 of *Lecture Notes in Computer Science*, pages 447–464. Springer, 2011.

[HS10] G. Hanrot and D. Stehlé. A complete worst-case analysis of Kannan's shortest lattice vector algorithm, 2010. Submitted.

[JS94] A. Joux and J. Stern. Lattice reduction: a toolbox for the cryptanalyst. *Journal of Cryptology*, 11:161–185, 1994.

[KÖ4] K. Königsberger. *Analysis.* Springer, Berlin, 2004.

[Kan83] 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, pages 193–206, New York, NY, USA, 1983. ACM.

[Kan87] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12:415–440, 1987.

[LaM91] B. A. LaMacchia. Basis reduction algorithms and subset sum problems. Technical report, SM thesis, Massachusetts Institute of Technology, 1991.

[Leh00] D.N. Lehmer. Asymptotic evaluation of certain totient sums. *American Journal of Mathematics*, 22(4):293–335, 1900.

[LJS90] 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):333–348, 1990.

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–536, 1982.

[LO85] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the ACM*, 32(1):229–246, 1985.

[Lod09] A. Lodi. Mixed integer programming computation. In *50 Years of Integer Programming 1958-2008*, pages 619–645. Springer, 2009.

[Maz10] G. Maze. Some inequalities related to the seysen measure of a lattice. *Linear Algebra and its Applications*, 433(810):1659 – 1665, 2010.

[Maz11] G. Maze. Natural density distribution of hermite normal forms of integer matrices. *Journal of Number Theory*, 131(12):2398 – 2408, 2011.

[MG02] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, 2002.

[MH78] R. C. Merkle and M. E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions On Information Theory*, 24(5):525–530, 1978.

[Mic01] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, 2001.

[MRW11] G. Maze, J. Rosenthal, and U. Wagner. Natural density of rectangular unimodular integer matrices. *Linear Algebra and its Applications*, 434(5):1319 – 1324, 2011.

[MW12] G. Maze and U. Wagner. A note on the weighted harmonic-geometric-arithmetic means inequalities. *Mathematical Inequalities & Applications*, 15(1):15–26, 2012.

[NS05] P.Q. Nguyen and D. Stehlé. Floating-point LLL revisited. In *Advances in Cryptology–Proceedings of EUROCRYPT '05*, volume 3494 of *Lecture Notes in Computer Science*, pages 215–233. Springer, Berlin, 2005.

[NS06] P. Q. Nguyen and D. Stehlé. LLL on the average. In *Algorithmic Number Theory*, Lecture Notes in Computer Science, pages 238–256. Springer, Berlin, 2006.

[NV08] P. Q. Nguyen and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2), 2008.

[NV10] P. Q. Nguyen and B. Vallée. *The LLL Algorithm: Survey and Applications*. Information Security and Cryptography. Springer, Berlin, 2010.

[Nym72] J.E. Nymann. On the probability that k positive integers are relatively prime. *Journal of Number Theory*, 4(5):469 – 473, 1972.

[Odl90] A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *Cryptology and computational number theory (Boulder, CO, '89)*, volume 42 of *Proceedings of Symposia in Applied Mathematics*, pages 75–88, Providence, RI, 1990. AMS.

[OG89] R. H. J. M. Otten and L. P. P. P. Van Ginneken. *The annealing algorithm*. Kluwer Academic Publishers, Boston, 1989.

[Qui76] D. Quillen. Projective modules over polynomial rings. *Inventiones mathematicae*, 36:167–172, 1976.

[Rog63] C. A. Rogers. Covering a sphere with spheres. *Mathematika*, 10(02):157–164, 1963.

[Sch87] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2-3):201–224, 1987.

[Sch94] C. P. Schnorr. Block reduced lattice bases and successive minima. *Combinatorics, Probability & Computing*, 3:507–522, 1994.

[SE94] C.P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(1-3):181–199, 1994.

[Sey93] M. Seysen. Simultaneous reduction of a lattice basis its reciprocal basis. *Combinatorica*, 13(3):363–376, 1993.

[Sus76] A.A. Suslin. Projective modules over polynomial rings are free. *Soviets Mathematics*, 4(17):1160–1164, 1976.

[TMK05] M. Taherzadeh, A. Mobasher, and A. K. Khandani. LLL lattice-basis reduction achieves the maximum diversity in MIMO systems. In *Proceedings of International Symposium on Information Theory, '05*, pages 1300–1304. IEEE, 2005.

[vdW03] B.L. van der Waerden. *Algebra: Volume 2*. Springer, New York, 2003.

[vEB81] P. van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, Math Inst., University of Amsterdam, Amsterdam, 1981.

[VG05] J.-L. Verger-Gaugry. Covering a ball with smaller equal balls in R^n. *Discrete & Computational Geometry*, 33(1):143–155, 2005.

[WM12] U. Wagner and G. Maze. Improvements in closest point search based on dual HKZ-bases. `arXiv:1201.5273`, 2012. Submitted.

[YP84] D. Youla and P. Pickel. The Quillen - Suslin theorem and the structure of n-dimensional elementary polynomial matrices. *IEEE Transactions on Circuits and Systems*, 31(6):513 – 518, 1984.

[ZAM08] W. Zhang, F. Arnold, and X. Ma. Fast communication: An analysis of Seysen's lattice reduction algorithm. *Signal Processing*, 88(10):2573–2577, 2008.