UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Algorithms for lattice problems with respect to general norms

Dipl.-Math.
Stefanie Naewe

November 10, 2011
(revised version)

A dissertation submitted to the
Department of Computer Science
University of Paderborn

for the degree of
Doktor der Naturwissenschaften
(doctor rerum naturalium)

accepted on the recommendation of

Prof. Dr. Johannes Blömer

University of Paderborn

Prof. Dr. Friedrich Eisenbrand

EPFL Lausanne

defended on

October 28, 2011

For My Parents

Acknowledgment

First of all, I wish to express special thanks to Prof. Dr. Johannes Blömer for being my thesis supervisor, particularly for his great support through the past years and the fruitful discussions about the ongoing progress in research.

The research presented in this thesis was supported by the Deutsche Forschungsgemeinschaft (DFG), grant BL 314/5 and Research Training Group GK-693 of the Paderborn Institute of Scientific Computation (PaSCo), and the Heinz Nixdorf Institute. I am grateful for the funding I received.

I have to thank my colleagues in the research group Dr. Marcel Ackermann, Jonas Gefele, Peter Günther, Claudia Jahn, Daniel Kuntze, Dr. Volker Krummel, and David Teusner for the very friendly and creative atmosphere.

I am also thankful to Andreas Cord-Landwehr, Christian Ikenmeyer, Eva Kuntze, Stefan Mengel, Holger Mense, and David Teusner for carefully proof-reading parts of my thesis and for giving valuable criticisms.

Last but not least, I would particular like to thank Holger Mense. This thesis would not have been completed without his continuous support and encouragement.

Abstract

Lattices are classical objects in the geometry of numbers. A lattice L is a discrete (abelian) subgroup of the n-dimensional vector space over the real numbers. Lattices have numerous applications ranging from number theory over computer algebra to optimization and cryptography.

In this thesis, we study the complexity of four classical problems from the geometry of numbers, the *shortest vector problem* (SVP), the *successive minima problem* (SMP), the *shortest independent vectors problem* (SIVP), and the *closest vector problem* (CVP). These problems can be defined for any norm on $\mathbb{R}^n$. The focus of this thesis is the algorithmic complexity of the four lattice problems described above with respect to arbitrary, especially non-Euclidean norms.

Extending and generalizing results of Ajtai et al. we present probabilistic single exponential time algorithms for all four lattice problems using single exponential space. The algorithms solve SVP and restricted versions of the other problems optimally using at most $(2^n \log_2(r))^{\mathcal{O}(1)}$ arithmetic operations, where n is the dimension of the vector space and r is an upper bound on the size of the input instance. Furthermore, the algorithms solve the general versions of SMP, SIVP, and CVP almost optimally, i.e., with approximation factor $1 + \epsilon$, where $0 < \epsilon < 3/2$. Here, the number of arithmetic operations of the algorithms is $((2 + 1/\epsilon)^n \log_2(r))^{\mathcal{O}(1)}$. While single exponential time algorithms that solve SVP optimally and CVP almost optimally were first presented in the seminal work of Ajtai et al., see [AKS01], [AKS02], the results for approximating SIVP and SMP improve upon previous results. Furthermore, Ajtai et al. describe their algorithm only for the Euclidean norm, whereas our algorithms work for any ℓ_p-norm, $1 \leq p \leq \infty$.

To obtain algorithms that solve SMP, SIVP, and CVP exactly with respect to arbitrary norms, we consider CVP in detail since there exist polynomial time reductions from SMP and SIVP to CVP which work for any norm, see [Mic08]. We will describe in this thesis deterministic polynomially space bounded algorithms for CVP for all ℓ_p-norms, $1 < p < \infty$, and all polyhedral norms, in particular for the ℓ_1-norm and the ℓ_∞-norm. For the running time we achieve the following results: For all ℓ_p-norms with $1 < p < \infty$ the number of arithmetic operations of the algorithm is $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r is an upper bound on the size of the CVP-instance and n is the dimension of the vector space. For polyhedral norms, we obtain an algorithm with running time $(s \cdot \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r and n are defined as above and s is the number of constraints defining the polytope. To the best of our knowledge this is the first result of this type. While there exist deterministic algorithms for CVP with respect to arbitrary norms using $n^{(4/3+o(1))n} \log_2(r)^{\mathcal{O}(1)}$ arithmetic operations, see [DPV11], [DV12], these algorithms do not run in polynomial space.

Zusammenfassung

Gitter sind klassische Objekte aus der Geometrie der Zahlen. Ein Gitter ist definiert als eine diskrete (abelsche) Untergruppe des $\mathbb{R}^n$. Gitter haben eine Vielzahl von Anwendungen, die von der Zahlentheorie über die Computeralgebra bis hin zur Optimierung und Kryptographie reichen.

Diese Dissertationsschrift beschäftigt sich mit der algorithmischen Komplexität von vier klassischen Problemen aus der Geometrie der Zahlen, dem *Problem des kürzesten Gittervektors*, dem *Problem der sukzessiven Minima*, dem *Problem der kürzesten linear unabhängigen Gittervektoren* sowie dem *Problem des nächsten Gittervektors.* Diese Probleme können bezüglich jeder beliebigen Norm auf dem $\mathbb{R}^n$ definiert werden. Der Schwerpunkt dieser Dissertation liegt auf der Untersuchung der algorithmischen Komplexität dieser oben erwähnten Gitterprobleme mit einem speziellen Fokus auf ihrer Lösbarkeit bezüglich allgemeiner, nicht euklidischer Normen.

Aufbauend auf Algorithmen von Ajtai, Kumar und Sivakumar ([AKS01], [AKS02]) für das Problem des kürzesten Gittervektors und das Problem des nächsten Gittervektors beschreiben wir in dieser Arbeit randomisierte Algorithmen mit einfach exponentieller Laufzeit für alle vier erwähnten Gitterprobleme. Diese Algorithmen lösen das Problem des kürzesten Gittervektors sowie wie eingeschränkte Varianten der anderen Gitterprobleme exakt. Dabei ist die Anzahl der arithmetischen Operationen beschränkt durch $(2^n \log_2(r))^{\mathcal{O}(1)}$, wobei n die Dimension des betrachteten Vektorraumes und r eine obere Schranke für die Eingabeinstanz ist. Für die allgemeinen Varianten des Problems der sukzessiven Minima, des Problems der kürzesten linear unabhängigen Gittervektoren sowie des Problems des nächsten Gittervektors beschreiben wir randomisierte Algorithmen mit einfach exponentieller Laufzeit, die diese Probleme mit Approximationsfaktor $1 + \epsilon$ für $0 < \epsilon < 3/2$ lösen. Die Anzahl der benötigten arithmetischen Operationen ist dabei beschränkt durch $((2 + 1/\epsilon)^n \log_2(r))^{\mathcal{O}(1)}$. Im Gegensatz zu den Algorithmen von Ajtai, Kumar und Sivakumar arbeiten alle von uns vorgestellen Algorithmen nicht nur für die euklidsche Norm sondern für allgemeine ℓ_p-Normen mit $1 \leq p \leq \infty$.

Um Algorithmen für das Problem der sukzessiven Minima, das Problem der kürzesten linear unabhängigen Gittervektoren sowie für das Problem des nächsten Gittervektors zu entwickeln, die diese Probleme exakt lösen, konzentrieren wir uns im zweiten Teil dieser Dissertationsschrift auf das Problem des nächsten Gittervektors. Dabei nutzen wir aus, dass sowohl das Problem der sukzessiven Minima als auch das Problem der kürzesten linear unabhängigen Gittervektoren polynomiell auf das Problem des nächsten Gittervektors reduzierbar sind, unabhängig von der entsprechenden Norm ([Mic08]).

Für das Problem des nächsten Gittervektors entwickeln wir in dieser Arbeit deterministische Algorithmen, die das Problem für alle ℓ_p-Normen mit $1 < p < \infty$ und alle Normen gegeben durch ein Polytop lösen, insbesondere auch für die ℓ_1-Norm und die ℓ_∞-Norm. Alle Algorithmen benötigen lediglich polynomiellen Platz. Allerdings ist die Anzahl der benötigten arithmetischen Operationen $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$, wenn man das Problem des nächsten Gittervektors bezüglich einer ℓ_p-Norm mit $1 < p < \infty$ löst. Dabei ist r eine obere Schranke für die Größe der Koeffizienten der Eingabeinstanz und n die Dimension des betrachteten Vektorraumes. Ist die Norm gegeben durch ein Polytop, so benötigt der Algorithmus zur Lösung des Problems des nächsten Gittervektors $(s \cdot \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$ arithmetische Operationen. Dabei sind die Parameter r und n wie oben definiert und s ist die Anzahl der Ungleichungen, die das Polytop definieren.
Zwar existieren deterministische Algorithmen, die das Problem des nächsten Gittervektors bezüglich allgemeiner Normen mit $n^{(4/3+o(1))n} \log_2(r)^{\mathcal{O}(1)}$ arithmetischen Operationen lösen ([DPV11], [DV12]), allerdings verwenden diese Algorithmen einfach exponentiell viel Platz.

Contents

1. Introduction 1

2. Norms and convex bodies 9

2.1. Equivalence between norms and convex bodies symmetric about the origin 9

2.1.1. Basic properties of convex sets and convex functions 9

2.1.2. Relation between norms and convex bodies symmetric about the origin 16

2.1.3. Algorithmic aspects of norms and convex bodies 19

2.2. Special convex bodies and the corresponding norms 22

2.2.1. Euclidean norms and ellipsoids 22

2.2.2. ℓ_p-norms and ℓ_p-balls with $1 \leq p \leq \infty$ 32

2.2.3. Polyhedral norms and polytopes 34

3. Lattices 39

3.1. Fundamentals about lattices 39

3.2. Minkowski's convex body theorem and successive minima 44

3.2.1. Minkowski's convex body theorem 44

3.2.2. Successive minima 46

3.2.3. Packing radius and covering radius 49

3.3. The dual lattice and transference bounds 50

3.3.1. Geometric representation of the dual lattice 51

3.3.2. Properties of the dual lattice 52

3.3.3. Transference bounds 54

4. Lattices: A complexity theoretic perspective 55

4.1. The lattice problems Svp, Smp, Sivp, and Cvp 55

4.2. Similarities and differences of the lattice problems 63

4.2.1. Orthogonal Projections 63

4.2.2. Number of solutions 68

4.3. Relation between lattice problems 73

4.3.1. The generalized shortest vector problem 74

4.3.2. The lattice membership problem 82

5. A randomized algorithm for the generalized shortest vector problem 91

5.1. A sampling procedure for approximate Gsvp 95

5.1.1. Preparations 96

5.1.2. Description of the sampling procedure 99

5.1.3. Analysis of the sampling procedure using a modified sampling procedure . . . 107
5.2. Using the sampling procedure for optimal solutions . . . 116
5.2.1. Description and analysis of the sampling procedure for optimal solutions . . . 116
5.2.2. Consequences for other lattice problems . . . 121
5.3. Discussion of the results . . . 126

6. A deterministic algorithm for the lattice membership problem 127
6.1. A general algorithm for the lattice membership problem . . . 130
6.1.1. The main idea of the lattice membership algorithm . . . 131
6.1.2. Description of the lattice membership algorithm . . . 132
6.1.3. A polynomially space bounded lattice membership algorithm . . . 136
6.2. A lattice membership algorithm for polytopes . . . 139
6.3. A lattice membership algorithm for ℓ_p-balls . . . 144
6.3.1. The class of general ℓ_p-balls . . . 144
6.3.2. Description and analysis of the algorithm . . . 145
6.4. An algorithm for computing a flatness direction . . . 151
6.4.1. A flatness algorithm for bounded convex sets . . . 152
6.4.2. A flatness algorithm for polytopes . . . 166
6.4.3. A flatness algorithm for ℓ_p-bodies . . . 169
6.5. Replacement procedure . . . 172
6.6. Discussion of the results . . . 183

7. Computation of approximate Löwner-John ellipsoids 185
7.1. The shallow cut ellipsoid method as a method to compute approximate Löwner-John ellipsoids . . . 189
7.1.1. Sufficient condition for an approximate Löwner-John ellipsoid . . . 191
7.1.2. Construction of a circumscribed ellipsoid . . . 199
7.1.3. Description and analysis of the rounding procedure for bounded convex sets . . . 208
7.2. A rounding method for ℓ_p-bodies . . . 217
7.2.1. Properties of ℓ_p-bodies . . . 217
7.2.2. Description and analysis of the algorithm . . . 224
7.3. A rounding method for polytopes . . . 228
7.3.1. Properties of polytopes . . . 228
7.3.2. Description and analysis of the algorithm . . . 230
7.4. Discussion of the results . . . 236

A. Appendix 239
A.0.1. Hadamard's inequality . . . 239
A.0.2. Chebyshev's inequality . . . 239
A.0.3. The Gamma function and Stirling's formula . . . 239

1. Introduction

Lattices are classical objects in the geometry of numbers, a mathematical theory established by Hermann Minkowski around 1900, see [Hil11]. A lattice L is a discrete (abelian) subgroup of the n-dimensional vector space over the real numbers. Each lattice has a basis that is a sequence of m elements of the lattice that generate the lattice as an abelian group. We call m the rank of the lattice.

Lattices establish the connection between discrete aspects of the Euclidean vector space $\mathbb{R}^n$, i.e., integer numbers, and elements from geometry, especially from the convex geometry. They have numerous applications ranging from number theory over computer algebra to optimization and cryptography.

In this thesis, we consider four classical problems from the geometry of numbers,

- the *shortest vector problem* (SVP), where we are given a lattice and want to find a shortest non-zero lattice vector,
- the *successive minima problem* (SMP), where we are given a lattice and want to successively compute linearly independent lattice vectors of minimal length,
- the *shortest independent vectors problem* (SIVP), where we are given a lattice and want to compute linearly independent lattice vectors with maximum length as short as possible, and
- the *closest vector problem* (CVP), where we are given a lattice together with some target vector from the vector space spanned by the lattice vectors and we want to compute the closest lattice vector to this target vector.

In the last 30 years, the complexity of these lattice problems has been studied intensively. It is known that all these problems are NP-hard and even hard to approximate, see for example [vEB81], [Ajt98], [ABSS93], [DKS98], [BS99], [Mic01], [DKRS03], [Kho05], [RR06], [HR07], and [Pei08].

The lattice problems SVP, SMP, SIVP, and CVP can be defined for any norm on $\mathbb{R}^n$. Thus, we stated them without referring to a specific norm. Often, they are considered with respect to the Euclidean norm. However, it is also common to consider these lattice problems with respect to other non-Euclidean norms, in particular the ℓ_∞-norm:

- Cryptosystems based on the knapsack problem can be broken if we can solve the shortest vector problem with respect to the ℓ_∞-norm, see [Rit96].

- If we are able to solve the closest vector problem with respect to the ℓ_∞-norm, we are able to solve the so-called hidden number problem which leads to attacks on the Digital Signature Algorithm (DSA), see [Ngu01] and [NS00].

- Up to now, the hardness of all lattice based cryptosystems is based on the hardness of lattice problems with respect to the Euclidean norm. But it seems that lattice problems in the Euclidean norm are easier than in any other norm, see [Pei08]. Hence, if we construct cryptosystems whose security is based on the hardness of certain lattice problems in a non-Euclidean norm, these cryptosystems are possibly harder to break.

- If we consider a polytope $\{x \in \mathbb{R}^m | Bx - t \leq \beta \cdot \mathbb{1}_n \text{ and } Bx - t \geq -\beta \cdot \mathbb{1}_n\}$ given by some nonsingular matrix $B \in \mathbb{R}^{n \times m}$, a vector $t \in \mathbb{R}^n$ and a radius $\beta > 0$, then the integer vectors in this polytope are characterized as the lattice vectors in the lattice $\mathcal{L}(B)$ whose distance to t with respect to the ℓ_∞-norm is at most β.

In this thesis, we study upper bounds on the complexity of the four lattice problems Svp, Smp, Sivp, and Cvp. Thereby, we concentrate on positive results, i.e., algorithms that solve these lattice problems either optimally or approximately. Furthermore we focus on their algorithmic complexity with respect to arbitrary norms.

Extending and generalizing results of Ajtai, Kumar, and Sivakumar we will present probabilistic single exponential time algorithms for all four lattice problems using single exponential space. The algorithms solve the shortest vector problem and restricted versions of the other problems optimally, using at most $(2^n \log_2(r))^{\mathcal{O}(1)}$ arithmetic operations, where n is the dimension of the vector space and r is an upper bound on the size of the input instance. Furthermore, the algorithms solve the general versions of Smp, Sivp, and Cvp almost optimally, i.e., with approximation factor $1 + \epsilon$ with $0 < \epsilon < 3/2$. Here, the number of arithmetic operations of the algorithms is $((2 + 1/\epsilon)^n \log_2(r))^{\mathcal{O}(1)}$ and the representation size of each number computed by the algorithm is polynomial in the representation size of the input.

While single exponential time algorithms that solve Svp optimally and Cvp almost optimally, were first presented in the seminal work of Ajtai, Kumar, and Sivakumar, see [AKS01], [AKS02], the results for Sivp and Smp improve upon previous results. Furthermore, Ajtai, Kumar, and Sivakumar describe their algorithms only for the Euclidean norm, our algorithms work for any so-called tractable norm, in particular for any ℓ_p-norm with $1 \leq p \leq \infty$.

While there exist deterministic single exponential time algorithms that solve all four lattice problems exactly in the Euclidean norm, for general ℓ_p-norms our approximation algorithms for Smp, Sivp, and Cvp are the best randomized algorithms. An exception is the algorithm of Eisenbrand, Hähnle, and Niemeier which is based on our algorithm for Cvp and approximates Cvp with approximation factor $1 + \epsilon$ using at most $((2 + \log_2(1/\epsilon))^n \log_2(r))^{\mathcal{O}(1)}$ arithmetic operations, see [EHN11].

To obtain algorithms that solve Smp, Sivp, and Cvp exactly with respect to arbitrary norms, we consider the closest vector problem in detail, since there exist polynomial time reductions from Smp and Sivp to Cvp that work for any norm and preserve the rank of the lattice, see [Mic08]. In this thesis, we will describe deterministic polynomially space bounded algorithms for the closest vector problem for all ℓ_p-norms, $1 < p < \infty$, and all polyhedral norms, in particular for the ℓ_1-norm and the ℓ_∞-norm. For the running time we achieve the following results: For all ℓ_p-norms with $1 < p < \infty$ the number of arithmetic operations of the algorithm is $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r is an upper bound on the size of the coefficients of the target vector and the lattice basis and n is the dimension of the vector space. For polyhedral norms, we obtain an algorithm using $(s \cdot \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$ arithmetic operations, where r and n are defined as above and s is the number of constraints defining the polytope. In particular, for the ℓ_1-norm and the ℓ_∞-norm, we obtain a deterministic algorithm for the closest vector problem which uses $\log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$ arithmetic operations. Since there are polynomial time reductions from Svp, Smp, and Sivp to Cvp that work for any norm, we obtain also deterministic algorithms for these problems.

For the shortest vector problem, this result is not really interesting since there exists already a deterministic polynomially space bounded algorithm that solves Svp in any ℓ_p-norm using at most $n^{(1+o(1))n} \log_2(r)^{\mathcal{O}(1)}$ arithmetic operations, see [Kan87b]. For the other three problems, to the best of our knowledge this is the first result of this type. While there exist algorithms for the closest vector problem with respect to arbitrary norms using $n^{(4/3+o(1))n} \log_2(r)^{\mathcal{O}(1)}$ arithmetic operations, see [DPV11], [DV12], these algorithms do not run in polynomial space.

Outline and main results

We briefly present the main results of this thesis and how it is organized.

Chapter 2 We start with a short introduction about arbitrary norms and their relation to convex bodies which are symmetric about the origin. Especially we show that every norm defines a unit ball which is a convex body symmetric about the origin and that every convex body symmetric about the origin can be used to define a norm. Furthermore, we consider some computational aspects of convex bodies that arise if we work with them in algorithms. At the end of this chapter, we introduce some special classes of convex bodies and consider the corresponding norms. These are ellipsoids, ℓ_p-balls with $1 \leq p \leq \infty$, and polytopes.

Chapter 3 In this chapter we give a short introduction into lattices and their connection to convex bodies. We define several fundamental concepts from the geometry of numbers and state the main important results. Particularly, we show how lattices interact with convex bodies, e.g., in Minkowski's convex body theorem which gives a sufficient criterion for the fact that a convex body contains a lattice vector.

Chapter 4 Whereas in Chapter 3 we considered lattices mainly from a pure mathematical point of view, in this chapter we focus on their computational aspects. We define the four classical lattice problems, Svp, Smp, Sivp, and Cvp and consider their complexity. We consider their similarities and differences as far as they are relevant for the development of algorithms for them. Particularly, we focus on the main difficulties that arise if we want to adapt an algorithm working for the Euclidean norm to an algorithm working for arbitrary norms.

At the end of this chapter, we make some preparations that we will use for the development of a unified algorithmic treatment for the four classical lattice problems.
Explicitly, we define a new lattice problem, the generalized shortest vector problem (Gsvp). This lattice problem is some kind of a generalization of the shortest vector problem: We are given some lattice L together with a subspace M of the $\mathbb{R}$-vector space $\operatorname{span}(L)$ spanned by the vectors in L. The goal is to compute a shortest lattice vector outside this subspace. Interestingly, the generalized shortest vector problem can also be seen as the generalization of the other three lattice problems. That means, there exist polynomial time reductions from the exact and approximate versions of Svp, Smp, Sivp, and Cvp to exact and approximate versions of the generalized shortest vector problem. The reductions work for any so-called tractable norm in particular for all ℓ_p-norms, $1 \leq p \leq \infty$.

Despite these results, it seems that the closest vector problem with respect to arbitrary norms is harder than the other lattice problems, since there exist polynomial time reductions from Svp, Smp, and Sivp to Cvp which work for any norm and preserve the approximation factor, see [Mic08]. Hence, we take a closer look on the closest vector problem and consider some kind of a geometric reformulation of the closest vector problem. We call this problem the lattice membership problem (Lmp): We are given a lattice L together with a bounded convex set $\mathcal{C}$ and the goal is to find a lattice vector in this convex set or to decide that the convex set does not contain a lattice vector. The lattice membership problem is a generalization of the integer programming feasibility problem from polyhedra to bounded convex sets. We show a polynomial time reduction from the closest vector problem to the lattice membership problem, which works for any so-called enumerable norm, in particular for any ℓ_p-norm, $1 \leq p \leq \infty$, and any polyhedral norm.

At the end of this chapter, we have two starting points for the development of lattice algorithms, the generalized shortest vector problem and the lattice membership problem as it is illustrated in Figure 1.1.

Parts of the results presented in this chapter are published in [BN07], [BN09], and [BN11] as a joint work with J. Blömer.

Chapter 5 In this chapter, we present a probabilistic single exponential time algorithm that approximates the generalized shortest vector problem with approximation factor

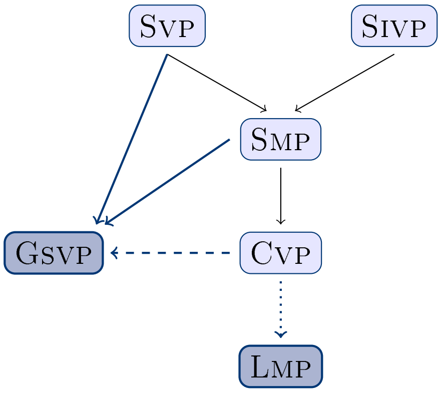

Figure 1.1.: **Relations among the lattice problems that will be used in this thesis.** Arrows indicate polynomial time reductions preserving the rank of the lattice and the approximation factor. The arrow from Cvp to Gsvp is marked dashed since the approximation factor is not exactly preserved by the reduction. The arrow from Cvp to Lmp is marked dotted since this reduction works only for the exact version of Cvp.

$1 + \epsilon$ for arbitrary $0 < \epsilon < 3/2$. Furthermore, we present an algorithm that solves for special restricted instances the generalized shortest vector problem exactly. These instances are characterized by the property that the number of $(1+\epsilon)$-approximate solutions is at most single exponential in the dimension.

The algorithms are based on the AKS-sampling technique developed by Ajtai, Kumar, and Sivakumar in 2001, see [AKS01]. It works for all so-called tractable norms, in particular for all ℓ_p-norms with $1 \leq p \leq \infty$. The number of arithmetic operations of the exact algorithm for Gsvp is $(2^n \log_2(r))^{\mathcal{O}(1)}$, where n is the dimension of the vector space and r is an upper bound on the size of the Gsvp-instance. The number of arithmetic operations of the approximate algorithm for Gsvp depends additionally on the approximation factor ϵ and is bounded from above by $((2 + (1/\epsilon))^n \log_2(r))^{\mathcal{O}(1)}$. The algorithm uses single exponential space but the representation size of each number computed by the algorithm is polynomial in the representation size of the input.

Using the polynomial time reduction from Chapter 4, we obtain also probabilistic single exponential time approximation algorithms for Svp, Smp, Sivp, and Cvp. Since for every instance of the shortest vector problem, the number of $(1+\epsilon)$-approximate solutions is at most $(2+\epsilon)^n$, we can use the exact algorithm for Gsvp to obtain a randomized single exponential time algorithm that solves Svp for all tractable norms exactly. We can show the same for the restricted versions of Smp, Sivp, and Cvp, and obtain also randomized single exponential time algorithms that solve these instances for all tractable norms exactly.

Parts of the results presented in this chapter are published in [BN07] and [BN09] as a joint work with J. Blömer.

Chapter 6 To obtain algorithms that solve Smp, Sivp, and Cvp in non-Euclidean norms exactly, we develop algorithms for the lattice membership problem. Since the lattice membership problem is a generalization of the integer programming feasibility problem, these algorithms are based on Lenstra's algorithm for integer programming, see [Len83]. Based on this algorithm, we present a general framework for algorithmic solutions for the lattice membership problem, which works for classes of bounded convex sets under the assumption that we have access to an algorithm that for each full-dimensional bounded convex set from this class computes a so-called approximate Löwner-John ellipsoid. For a full-dimensional bounded convex set $\mathcal{C}$, an approximate Löwner-John ellipsoid is an ellipsoid which is contained in $\mathcal{C}$ and the approximation factor is the factor which we need to scale the ellipsoid with such that the scaled ellipsoid contains the convex set. Under the assumption that we are able to compute such an approximate Löwner-John ellipsoid for polytopes and generalizations of ℓ_p-balls, we obtain a deterministic polynomially space bounded algorithm for the lattice membership problem where the convex sets are polytopes or ℓ_p-balls. In Chapter 7, we will show that this assumption is true, i.e., we will show that there exists algorithms that compute approximate Löwner-John ellipsoids for polytopes and generalizations of ℓ_p-balls.

Using the deterministic polynomially space bounded algorithm for the lattice membership problem together with the polynomial time reduction from the closest vector problem to the lattice membership problem, we obtain a deterministic polynomially space bounded algorithm for the closest vector problem that works for all ℓ_p-norms, $1 \leq p \leq \infty$, and all polyhedral norms. The number of arithmetic operations of this algorithm is $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$ if we consider an ℓ_p-norm with $1 < p < \infty$, where n is the dimension of the lattice and r is an upper bound on the size of the Cvp-instance. If we consider the closest vector problem with respect to a polyhedral norm, we obtain an algorithm where the number of arithmetic operations is $(s \cdot \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$, where n and r are defined as above and s is the number of constraints defining the polytope.

Parts of the results presented in this chapter are published in [BN11] as a joint work with J. Blömer.

Chapter 7 As mentioned before, to realize the lattice membership algorithm presented in Chapter 6 for a concrete class of bounded convex sets, we need access to an algorithm that computes an approximate Löwner-John ellipsoid for the convex sets from this class.

For the class of full-dimensional polytopes, there exists such an algorithm. Extending a method from Lenstra, Goffin described in 1984 a polynomial time algorithm that computes a $2n$-approximate Löwner-John ellipsoid for a full-dimensional polytope in $\mathbb{R}^n$, see [Gof84]. This algorithm is based on a variant of the ellipsoid method developed by Shor, Yudin and Nemirovskii.

The second class we consider are generalizations of ℓ_p-balls, so called ℓ_p-bodies. These ℓ_p-bodies are full-dimensional bounded convex sets which are defined as the image of an ℓ_p-ball under a bijective affine transformation intersected with hyperplanes orthogonal to the unit vectors. In this chapter, we present an algorithm that computes a $2n$-approximate Löwner-John ellipsoid for an ℓ_p-body of dimension n. The algorithm is a concrete realization of a general algorithmic framework that computes for a given full-dimensional bounded convex set an approximate Löwner-John ellipsoid with approximation factor $c \cdot n$ for some constant $c > 1$. This algorithmic framework is a variant of a polynomial time algorithm due to Grötschel, Lovász and Schrijver, which is based on the ellipsoid method and computes a $\sqrt{n}(n+1)$-approximate Löwner-John ellipsoid of a convex body, combined with some ideas of Kochol, Hildebrand, and Köppe. One can show that the approximation factor achieved by our algorithm is almost optimal. The number of arithmetic operations of the algorithm is single exponential in the dimension.

The results presented in this chapter complete the description of the algorithms for the lattice membership problem presented in Chapter 6.

Parts of the results presented in this chapter are published in [BN11] as a joint work with J. Blömer.

2. Norms and convex bodies

In this thesis we consider lattices, which are discrete objects in the vector space $\mathbb{R}^n$, where $n \in \mathbb{N}$. Thereby, we focus on certain computational problems related to lattices, so-called lattice problems. These problems are often defined with respect to some norm on the vector space $\mathbb{R}^n$. Thus, before we give a formal definition of lattices and state their main properties, we give in this chapter a short introduction into convexity as far as it is needed for the understanding of this thesis.

In particular we show that there is an equivalence between convex bodies symmetric about the origin and norms: Every norm on $\mathbb{R}^n$ can be used to define a convex body in $\mathbb{R}^n$ which is symmetric about the origin. Conversely, every convex body in $\mathbb{R}^n$ which is symmetric about the origin defines a norm on $\mathbb{R}^n$. This equivalence enables us to use results of convex geometry in functional analysis and vice versa.

At the end of this chapter, we consider some special classes of norms respectively convex bodies in detail: The Euclidean norm and general Euclidean norms together with ellipsoids, ℓ_p-norms and the corresponding ℓ_p-balls with $1 \leq p \leq \infty$, and finally polyhedral norms and polytopes.

2.1. Equivalence between norms and convex bodies symmetric about the origin

We start with some basics about convexity, especially convex sets and convex functions. Most of the results in this section appear without a proof. They can be found together with more details on this topic in [BV09], [Web94], and [Roc70].

2.1.1. Basic properties of convex sets and convex functions

Convex sets

Geometrically, a set $\mathcal{C} \subseteq \mathbb{R}^n$ is convex if for any two vectors $x, y \in \mathcal{C}$ the line segment between them lies in $\mathcal{C}$. The set $\mathcal{C}$ is strictly convex if for any vectors $x, y \in \mathcal{C}$, every point on the line segment between them lies in the interior of $\mathcal{C}$, $\text{int}(\mathcal{C})$.

Definition 2.1.1. *((Strictly) convex set)*
A set $\mathcal{C} \subseteq \mathbb{R}^n$ is convex *if for any two vectors $x, y \in \mathbb{R}^n$ and $0 \leq \theta \leq 1$, we have*

$$\theta \cdot x + (1 - \theta) \cdot y \in \mathcal{C}.$$

The set $\mathcal{C}$ is strictly convex *if for two vectors $x, y \in \mathbb{R}^n$, $x \neq y$, and $0 < \theta < 1$ we have that*

$$\theta \cdot x + (1-\theta) \cdot y \in \operatorname{int}(\mathcal{C}).$$

Important convex sets in $\mathbb{R}^n$ are the empty set $\emptyset$, any single vector $\{x\}$ with $x \in \mathbb{R}^n$, and the whole vector space $\mathbb{R}^n$.

Given k vectors $x_1, \dots, x_k \in \mathbb{R}^n$, the *convex hull* of these vectors is defined as the smallest convex set which contains these vectors. It is denoted by $\operatorname{conv}(x_1, \dots, x_k)$. We have

$$\operatorname{conv}(x_1, \dots, x_k) = \left\{ \sum_{i=1}^{k} \theta_i x_i \,\middle|\, \theta_i \geq 0 \text{ satisfying } \sum_{i=1}^{k} \theta_i = 1 \right\}.$$

The *dimension* of a non-empty affine subspace is defined as the dimension of the subspace parallel to it. By convention, the dimension of $\emptyset$ is -1. We define the *dimension of a set* as the dimension of the smallest affine subspace containing it. A convex set which is full-dimensional and compact is called a convex body.

Definition 2.1.2. *(Convex Body)*
A compact convex set $\mathcal{C} \subseteq \mathbb{R}^n$ with non-empty interior is called a convex body.

There are some basic operations for subsets of $\mathbb{R}^n$ that preserve convexity. If we apply one of these operations to convex sets, the result is also a convex set. These operations allow the construction of new convex sets from other convex sets.

- The translation of a (convex) set $\mathcal{C} \subseteq \mathbb{R}^n$ by a vector $t \in \mathbb{R}^n$ is the set

  $$\mathcal{C} + t := \{x + t | x \in \mathcal{C}\}.$$

  If $\mathcal{C}$ is (strictly) convex, $\mathcal{C} + t$ is (strictly) convex.

- For two (convex) sets $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{R}^n$ their Minkowski sum is defined as

  $$\mathcal{C}_1 + \mathcal{C}_2 := \{x + y | x \in \mathcal{C}_1, y \in \mathcal{C}_2\}.$$

  If $\mathcal{C}_1$ and $\mathcal{C}_2$ are (strictly) convex, their Minkowski sum $\mathcal{C}_1 + \mathcal{C}_2$ is also a (strictly) convex set.

- For a scalar $\theta \in \mathbb{R}$ we define the scaling of $\mathcal{C}$ by the factor θ as the set

  $$\theta \cdot \mathcal{C} := \{\theta \cdot x | x \in \mathcal{C}\}.$$

  One can show that if $\mathcal{C}$ is (strictly) convex, the set $\theta \cdot \mathcal{C}$ is also a (strictly) convex set. We observe that for positive real numbers $\theta_1, \theta_2 > 0$ we have

  $$(\theta_1 + \theta_2) \cdot \mathcal{C} = \theta_1 \cdot \mathcal{C} + \theta_2 \cdot \mathcal{C}.$$

- Another important observation is that convexity is preserved under intersection. If $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{R}^n$ are (strictly) convex sets, their intersection $\mathcal{C}_1 \cap \mathcal{C}_2$ is also a (strictly) convex set.

Furthermore, convexity is preserved by bijective affine transformation. If $f : \mathbb{R}^n \to \mathbb{R}^n$ is a mapping of the form $x \mapsto Q \cdot x + q$, where $Q \in \mathbb{R}^{n \times n}$ is nonsingular and $q \in \mathbb{R}^n$, then the image $f(\mathcal{C})$ of a convex set $\mathcal{C} \subseteq \mathbb{R}^n$ under this mapping f is also a convex set.

One of the main characterizing properties of a set is its volume. The following result is fundamental for the computation of volumes of sets.

Lemma 2.1.3. *Let $\mathcal{S} \subseteq \mathbb{R}^n$ be a measurable set. Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be an affine transformation given by $x \mapsto Q \cdot x + q$ for $x \in \mathbb{R}^n$, where $Q \in \mathbb{R}^{n \times n}$, $q \in \mathbb{R}^n$. Then*

$$\mathrm{vol}_n(T(\mathcal{S})) = |\det(Q)| \cdot \mathrm{vol}_n(\mathcal{S}).$$

Particularly, we obtain that for $\theta > 0$ that the set $\theta \cdot \mathcal{S}$ has volume

$$\mathrm{vol}_n(\theta \cdot \mathcal{S}) = \theta^n \cdot \mathrm{vol}_n(\mathcal{S}).$$

Separating and supporting hyperplanes

We now describe an important idea in convexity which has a great influence on the algorithmic use of convexity.

Every vector $d \in \mathbb{R}^n \backslash \{0\}$ defines a *family of affine hyperplanes* in $\mathbb{R}^n$ by

$$H_{k,d} := \{x \in \mathbb{R}^n | \langle d, x \rangle = k\},$$

where $k \in \mathbb{R}$. The set $H_{0,d} = \{x \in \mathbb{R}^n | \langle x, d \rangle = 0\}$ is called a *hyperplane.* Every hyperplane is a subspace of $\mathbb{R}^n$ of dimension $n-1$. Here $\langle \cdot, \cdot \rangle$ denotes the Euclidean scalar product, i.e., $\langle x, y \rangle = \sum_{i=1}^n x_i \cdot y_i$ for $x, y \in \mathbb{R}^n$.

From the analytical point of view, an affine hyperplane is the solution set of a non-trivial linear equation. From the geometrical point of view, an affine hyperplane is the set of all vectors which have a constant scalar product with some given vector d, called the normal vector. In this case, the constant k defines the translation of the hyperplane from the origin. If $x_0 \in \mathbb{R}^n$ is an arbitrary vector in the affine hyperplane $H_{k,d}$, i.e., if $\langle d, x_0 \rangle = k$, then we have $H_{k,d} = \{x \in \mathbb{R}^n | \langle d, x - x_0 \rangle = 0\}$.

Any affine hyperplane separates the vector space $\mathbb{R}^n$ in two halfspaces. A (closed) halfspace is the set of all vectors of the form $\{x \in \mathbb{R}^n | \langle x, d \rangle \leq k\}$ with $d \in \mathbb{R}^n \backslash \{0\}$ and $k \in \mathbb{R}$, that means the solution set of a non-trivial linear inequality. Obviously, halfspaces and (affine) hyperplanes are convex sets.

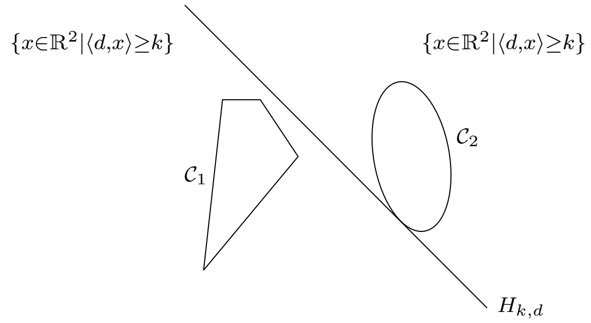
$\{x\in\mathbb{R}^2|\langle d,x\rangle\geq k\}$
$\{x\in\mathbb{R}^2|\langle d,x\rangle\geq k\}$
$\mathcal{C}_2$
$\mathcal{C}_1$
$H_{k,d}$

Figure 2.1.: **Separating hyperplanes.** The affine hyperplane $H_{k,d}$ separates the disjoint sets $\mathcal{C}_1$ and $\mathcal{C}_2$.

Halfspaces can be used to give an alternative characterization of convex sets. Every closed convex set $\mathcal{C} \subseteq \mathbb{R}^n$ can be represent as the intersection of all halfspaces containing it,

$$\mathcal{C} = \bigcap \{H^- | H^- \subseteq \mathbb{R}^n \text{ halfspace with } \mathcal{C} \subseteq H^-\}.$$

If a convex set can be represented as the intersection of finitely many halfspaces, it is called a *polyhedron*.

The following result is fundamental in the idea of convexity. It states that for each two disjoint convex sets there exists an affine hyperplane that separates them, as it is illustrated in Figure 2.1. A proof of the following theorem can be found for example in [BV09].

Theorem 2.1.4. *(Separating hyperplane theorem)*
Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{R}^n$ be two disjoint convex sets, i.e. $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$. Then there exists an affine hyperplane $\{x \in \mathbb{R}^n | \langle d, x\rangle = k\}$ given by a vector $d \in \mathbb{R}^n \backslash \{0\}$ and a number $k \in \mathbb{R}$ which separates $\mathcal{C}_1$ and $\mathcal{C}_2$. That means for all $x \in \mathcal{C}_1$ we have $\langle d, x\rangle \leq k$ and for all $x \in \mathcal{C}_2$ we have $\langle d, x\rangle \geq k$.

If we consider a convex set $\mathcal{C} \subseteq \mathbb{R}^n$ together with a vector $v \in \mathbb{R}^n$ which is not contained in $\mathcal{C}$, then the separating hyperplane theorem guarantees that there exists an affine hyperplane which separates $\mathcal{C}$ and v. That means, there exists a vector $d \in \mathbb{R}^n \backslash \{0\}$ such that

$$\langle d, x\rangle \leq \langle d, v\rangle \text{ for all } x \in \mathcal{C}.$$

We call such an affine hyperplane a *separating hyperplane*. An affine hyperplane $H_{k,d}$ *strictly separates* the vector v from the set $\mathcal{C}$ if $\langle d, v\rangle < k < \langle d, x\rangle$ for all $x \in \mathcal{C}$. In general, it is not guaranteed that for two convex sets, there exists an affine hyperplane that strictly separates them. But in some special cases, for example if one of the convex sets consists of a single vector and the other convex set is closed, one can show that there exists an affine hyperplane that strictly separates v from $\mathcal{C}$.

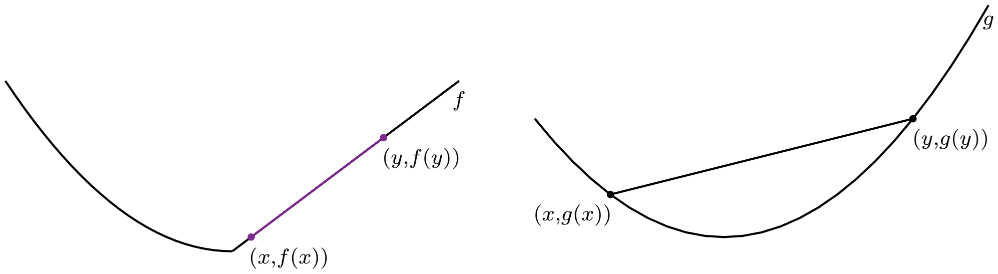

Figure 2.2.: **Convex and strictly convex functions.** The function f is convex but not strictly convex. The line segment between the points $(x, f(x))$ and $(y, f(y))$ lies on the graph of f. The function g is strictly convex since for all x, y the line segment between the points $(x, f(x))$ and $(y, f(y))$ lies above the graph.

Lemma 2.1.5. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a closed convex set and $v \notin \mathcal{C}$. Then there exists a vector $d \in \mathbb{R}^n$ such that $\langle d, v\rangle < \langle d, x\rangle$ for all $x \in \mathcal{C}$.*

If we are given some vector $v \in \mathbb{R}^n$ on the boundary of some set $\mathcal{C} \subseteq \mathbb{R}^n$ and there exists a vector $d \in \mathbb{R}^n \backslash \{0\}$ such that $\langle d, x\rangle \leq \langle d, v\rangle$ for all $x \in \mathcal{C}$, the affine hyperplane $H_{\langle d,v\rangle,d} = \{x \in \mathbb{R}^n | \langle d, x\rangle = \langle d, v\rangle\}$ is called a *supporting hyperplane* to the set $\mathcal{C}$ at the vector v. The geometric interpretation of this situation is that the affine hyperplane $H_{\langle d,v\rangle,d}$ is a tangent to the set $\mathcal{C}$ at the vector v and the halfspace $\{x \in \mathbb{R}^n | \langle x, d\rangle = \langle d, v\rangle\}$ contains $\mathcal{C}$. Based on the separating hyperplane theorem is can be shown that for every convex set $\mathcal{C} \subseteq \mathbb{R}^n$ and every vector v on its boundary there exists an affine hyperplane given by a vector $d \in \mathbb{R}^n \backslash \{0\}$ which supports the set $\mathcal{C}$ at the vector v.

Convex functions

Geometrically, a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex on $\mathbb{R}^n$ if for all $x, y \in \mathbb{R}^n$ the line segment between $(x, f(x))$ and $(y, f(y))$ lies above the graph of f.

Definition 2.1.6. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is* convex *if its domain,* dom(f)*, is a convex set and if for all $x, y \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ with $0 \leq \theta \leq 1$ we have*

$$f(\theta \cdot x + (1 - \theta) \cdot y) \leq \theta \cdot f(x) + (1 - \theta) \cdot f(y).$$

The function f is strictly convex *on $\mathbb{R}^n$ if for all $x, y \in \mathbb{R}^n$ linearly independent and $0 < \theta < 1$ we have*

$$f(\theta \cdot x + (1 - \theta) \cdot y) < \theta \cdot f(x) + (1 - \theta) \cdot f(y).$$

The difference between convex functions and strictly convex functions is illustrated in Figure 2.2.

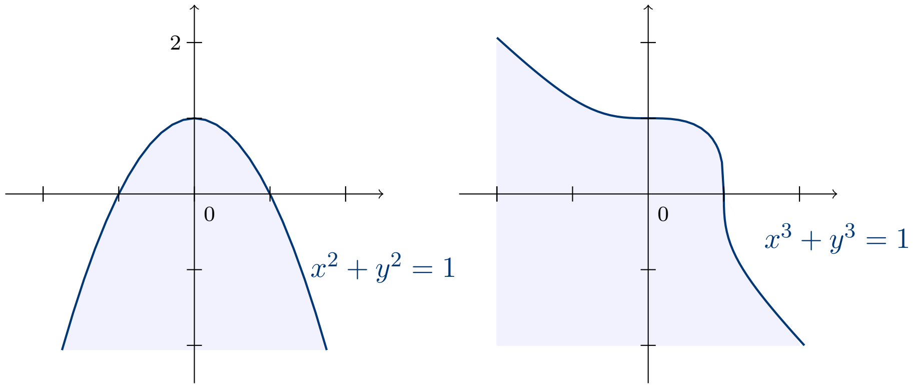

Figure 2.3.: **Quasiconvex functions.** The left picture shows the 1-sublevel set of the function $(x, y) \mapsto x^2 + y^2$, which is quasiconvex. The right picture shows the 1-sublevel set of the function $(x, y) \mapsto x^3 + y^3$, which is not quasiconvex.

This shows that we can characterize the convexity of a function geometrically via its graph. We can formalize this notion as follows: A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if its epigraph

$$\text{epi}(f) = \{(x, t) | x \in \mathbb{R}^n \text{ with } f(x) \leq t\} \subseteq \mathbb{R}^{n+1}$$

is a convex set.

For a function $f : \mathbb{R}^n \to \mathbb{R}$ and a parameter $\alpha \in \mathbb{R}$, an *α-sublevel set* of f is defined as the set

$$\mathcal{C}_\alpha := \{x \in \text{dom}(f) | f(x) < \alpha\}.$$

For all $\alpha \in \mathbb{R}$ the α-sublevel set of f is a convex set if f is a convex function. The converse is not true: For example, the function $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto -e^x$ is not convex although all its sublevel sets are convex. Such a function is called quasiconvex.

Definition 2.1.7. *(Quasiconvex function)*
A function $f : \mathbb{R}^n \to \mathbb{R}$ is called quasiconvex*, if all α-sublevel sets $\mathcal{C}_\alpha$, $\alpha \in \mathbb{R}$ are convex sets.*

An example for a function which is not quasiconvex is the function $f : \mathbb{R}^n \to \mathbb{R}$, $x \mapsto \sum_{i=1}^n x_i^3$, see Figure 2.3 for an illustration.

For differentiable functions $f : \mathbb{R}^n \to \mathbb{R}$ there is another possibility to characterize them as convex functions using its first-order Taylor expansion. For a vector $x \in \mathbb{R}^n$ the first-order Taylor expansion of f is given by the function $y \mapsto f(x) + \nabla f(x)^T (y - x)$, where $\nabla f(x) \in \mathbb{R}^n$ denotes the gradient of f at the vector x. The important property of

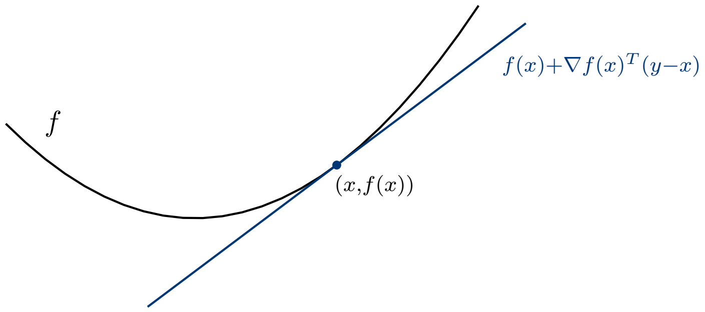

Figure 2.4.: **First-order convexity condition.** The function $f : \mathbb{R} \to \mathbb{R}$ is convex and differentiable. It's first-order Taylor expansion is a global underestimator of f.

a convex function is that for all $x \in \mathrm{dom}(f)$ the first-order Taylor expansion is a global underestimator of the function f and vice versa. This criterion is known as the first-order convexity condition. Its geometric idea is illustrated in Figure 2.4.

Lemma 2.1.8. *(First-order convexity condition)*
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. Then f is convex if and only if its domain $\mathrm{dom}(f)$ is convex and if

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) \text{ for all } x, y \in \mathrm{dom}(f).$$

Also strict convexity can be characterized using the first-order convexity condition. A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex if and only if its domain $\mathrm{dom}(f)$ is convex and if $f(y) > f(x) + \nabla f(x)^T(y-x)$ for all $x, y \in \mathrm{dom}(f)$, $x \neq y$.

Norms

In this thesis, we work with special convex functions called norms. In general, a norm on $\mathbb{R}^n$ is defined by a function $p : \mathbb{R}^n \to \mathbb{R}$, which satisfies certain properties.

Definition 2.1.9. *(Norm)*
A function $p : \mathbb{R}^n \to \mathbb{R}$ is called a norm *on $\mathbb{R}^n$ if it satisfies the following properties:*

- **(Positivity)** *For all $x \in \mathbb{R}^n$ it holds that $p(x) \geq 0$. Furthermore, we have $p(x) = 0$ if and only if $x = 0$.*
- **(Absolute Homogenity)** *We have $p(\lambda \cdot x) = |\lambda| \cdot p(x)$ for all $x \in \mathbb{R}^n, \lambda \in \mathbb{R}$.*
- **(Subadditivity)** *We have $p(x+y) \leq p(x) + p(y)$ for all $x, y \in \mathbb{R}^n$.*

The last property is also known as the triangle inequality. It follows directly from the positivity and the absolute homogenity that every norm is a convex function. Furthermore we observe that a norm is strictly convex if and only if for all $x, y \in \mathbb{R}^n$ linearly independent we have $\|x+y\| < \|x\| + \|y\|$.

The commonly used norm is the Euclidean norm, which is defined as

$$\|\cdot\|_2 : \mathbb{R}^n \to \mathbb{R},\ x \mapsto \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}.$$

Sometimes this norm is also called the ℓ_2-norm. For a vector $x \in \mathbb{R}^n$ its Euclidean norm is that what is usually associated with the length of a vector. The overall approach of norms as defined in Definition 2.1.9 generalizes this notion of the length of a vector.

2.1.2. Relation between norms and convex bodies symmetric about the origin

After this short introduction, we now focus on the relation between norms and convex bodies symmetric about the origin. As we already mentioned, for every convex function and every parameter $\alpha \in \mathbb{R}$ the corresponding α-sublevel set is a convex set. For a norm $\|\cdot\|$ on $\mathbb{R}^n$ we denote this sublevel sets by $B_n^{(\|\cdot\|)}(0,\alpha) := \{y \in \mathbb{R}^n | \|y\| < \alpha\}$. If we translate this sets by a vector $x \in \mathbb{R}^n$, we obtain

$$B_n^{(\|\cdot\|)}(x,\alpha) := \big\{y \in \mathbb{R}^n \mid \|y-x\| < \alpha\big\}.$$

We call this convex set the *ball generated by the norm* $\|\cdot\|$ with center $x \in \mathbb{R}^n$ and radius $\alpha > 0$. By $\bar{B}_n^{(\|\cdot\|)}(x,\alpha)$ we denote the corresponding closed ball,

$$\bar{B}_n^{(\|\cdot\|)}(x,\alpha) := \big\{y \in \mathbb{R}^n \mid \|y-x\| \leq \alpha\big\}.$$

The ball $\bar{B}_n^{(\|\cdot\|)}(0,1)$ is called the *unit ball of the norm* $\|\cdot\|$. The ball corresponding to the Euclidean norm is denoted by $\bar{B}_n^{(2)}(x,\alpha)$. The volume $\mathrm{vol}_n(B_n^{(\|\cdot\|)}(x,\alpha))$ of the ball $B_n^{(\|\cdot\|)}(x,\alpha)$ satisfies the following condition,

$$\mathrm{vol}_n(B_n^{(\|\cdot\|)}(x,\theta\cdot\alpha)) = \theta^n \cdot \mathrm{vol}_n(B_n^{(\|\cdot\|)}(x,\alpha)) \tag{2.1}$$

for all $\theta > 0$. This result follows directly from Lemma 2.1.3 and will be used several times in this thesis.

For every norm the corresponding unit ball is a convex body which is symmetric about the origin. The main part to prove this is to show that the unit ball of a norm is compact and has non-empty interior. This is based on the observation that every norm is a continuous mapping.

Lemma 2.1.10. *Let* $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}^{\geq 0}$ *be a norm on* $\mathbb{R}^n$*. Then* $\|\cdot\|$ *is continuous.*

Proof. We consider the standard basis of the vector space $\mathbb{R}^n$ given by the vectors $e_1, \ldots, e_n \in \mathbb{R}^n$. That means for every vector $x \in \mathbb{R}^n$ with $x = (x_1, \ldots, x_n)^T$ we have $x = \sum_{i=1}^{n} x_i e_i$. We set

$$\gamma := \max\{\|e_i\| | 1 \leq i \leq n\}.$$

Then it follows from the subadditivity of the norm that

$$\|x\| = \|\sum_{i=1}^{n} x_i e_i\| \leq \sum_{i=1}^{n} |x_i| \cdot \|e_i\| \leq \gamma \cdot \sum_{i=1}^{n} |x_i|.$$

For any two vectors $x, y \in \mathbb{R}^n$ we obtain from the triangle inequality that

$$\begin{aligned} \|x\| &= \|y + (x - y)\| \leq \|y\| + \|x - y\| \text{ and} \\ \|y\| &= \|x + (y - x)\| \leq \|x\| + \|y - x\|. \end{aligned}$$

This shows that

$$\big|\|x\| - \|y\|\big| \leq \|x - y\| \leq \gamma \cdot \sum_{i=1}^{n} |x_i - y_i|$$

from which it follows that $\| \cdot \|$ is continuous. □

Using this, we can show that for every norm the corresponding unit ball is a convex body.

Proposition 2.1.11. *Let $\| \cdot \|$ be a norm on $\mathbb{R}^n$. Then the set*

$$\bar{B}_n^{(\|\cdot\|)}(0,1) := \{x \in \mathbb{R}^n |\ \|x\| \leq 1\}.$$

is a convex body symmetric about the origin.

Proof. To prove the convexity of the set $\bar{B}_n^{(\|\cdot\|)}(0,1)$ we consider two vectors $x, y \in \bar{B}_n^{(\|\cdot\|)}(0,1)$ together with a parameter θ satisfying $0 < \theta \leq 1$. Since $\| \cdot \|$ is a norm on $\mathbb{R}^n$, it holds that

$$\|\theta \cdot x + (1 - \theta) \cdot y\| \leq |\theta| \cdot \|x\| + |1 - \theta| \cdot \|y\| \leq \theta + (1 - \theta) = 1,$$

where we use that $\|x\| \leq 1$ and $\|y\| \leq 1$. The symmetry of the set $\bar{B}_n^{(\|\cdot\|)}(0,1)$ follows from the absolute homogenity of the norm. We have $\|x\| = \| - x\|$ for all $x \in \mathbb{R}^n$. This shows that $\bar{B}_n^{(\|\cdot\|)}(0,1)$ is a convex set symmetric about the origin.
It remains to show that the set is compact, i.e., closed and bounded, and that it has non-empty interior. To show this, we use that every norm is a continuous mapping as we have seen in Lemma 2.1.10. We consider the closed set

$$\big\{\lambda \in \mathbb{R} \mid |\lambda| \leq 1\big\}$$

which is the image of the unit ball $\bar{B}_n^{(\|\cdot\|)}(0,1)$ under the mapping $\| \cdot \|$,

$$\| \cdot \| : \bar{B}_n^{(\|\cdot\|)}(0,1) \mapsto \big\{\lambda \in \mathbb{R} \mid |\lambda| \leq 1\big\}.$$

That means, the set $\bar{B}_n^{(\|\cdot\|)}(0,1)$ is the preimage of a closed set under a continuous mapping which shows that $\bar{B}_n^{(\|\cdot\|)}(0,1)$ is also closed.
To prove that $\bar{B}_n^{(\|\cdot\|)}(0,1)$ is bounded we consider the unit sphere of the Euclidean norm,

$$\mathbb{S}^{n-1} := \big\{x \in \mathbb{R}^n | \ \|x\|_2 = 1\big\}.$$

Obviously, we have $\|x\| > 0$ for all $x \in \mathbb{S}^{n-1}$. Since $\|\cdot\|$ is continuous, there exists a $\delta > 0$ such that $\|x\| > \delta$ for all $x \in \mathbb{S}^{n-1}$. Hence, it follows that for all $x \in \bar{B}_n^{(\|\cdot\|)}(0,1)$ we have

$$\frac{1}{\|x\|_2} \geq \frac{\|x\|}{\|x\|_2} = \Big\|\frac{x}{\|x\|_2}\Big\| > \delta,$$

where the last inequality follows from $x/\|x\|_2 \in \mathbb{S}^{n-1}$. This shows that $\|x\|_2 < 1/\delta$ for all $x \in \bar{B}_n^{(\|\cdot\|)}(0,1)$ and that $\bar{B}_n^{(\|\cdot\|)}(0,1)$ is compact.
Finally, we show that the origin is an interior point of the unit ball $\bar{B}_n^{(\|\cdot\|)}(0,1)$. Since $\|\cdot\|$ is continuous, there exists a $\beta > 0$ such that $\|x\| \leq \beta$ for all $x \in \mathbb{S}^{n-1}$. Hence, for all $x \in \mathbb{R}^n$ with $\|x\|_2 \leq 1/\beta$ we obtain that

$$\|x\| = \frac{\|x\|}{\|x\|_2} \cdot \|x\|_2 < \Big\|\frac{x}{\|x\|_2}\Big\| \cdot \frac{1}{\beta} \leq \beta \cdot \frac{1}{\beta} = 1.$$

The last inequality is due to the fact that $x/\|x\|_2 \in \mathbb{S}^{n-1}$. □

To show that every convex body symmetric about the origin defines a norm, we use the so-called Minkowski function of a set. For a given set $\mathcal{C} \subseteq \mathbb{R}^n$ and a vector $x \in \mathbb{R}^n$ the value of the Minkowski function of $\mathcal{C}$ at x is that minimal positive real number θ by which we need to scale the set such that x is contained in the set $\theta \cdot \mathcal{C}$. The Minkowski function of a convex body is illustrated in Figure 2.5.

Definition 2.1.12. *(Minkowski function)*
Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a set. The Minkowski function of $\mathcal{C}$ is defined as the function

$$F_{\mathcal{C}} : \mathbb{R}^n \to [0, \infty], \ x \mapsto \inf\{\theta > 0 | x \in \theta\mathcal{C}\}.$$

If the set $\mathcal{C}$ is closed, $F_{\mathcal{C}}(x)$ is the positive real number θ such that x is contained on the boundary of $\theta \cdot x$. Additionally, we observe that a vector $x \in \theta \cdot \mathcal{C}$ if and only if the vector $\theta^{-1}x \in \mathcal{C}$.

Proposition 2.1.13. *Let $\mathcal{C} \subsetneq \mathbb{R}^n$ be a convex body symmetric about the origin. Then the Minkowski function*

$$F_{\mathcal{C}} : \mathbb{R}^n \to \mathbb{R}, \ x \mapsto \inf\{\rho > 0 | x \in \rho \cdot \mathcal{C}\}$$

is a norm on $\mathbb{R}^n$.

Often, the norm defined by a convex body $\mathcal{C}$ symmetric about the origin is denoted by $\|\cdot\|_{\mathcal{C}}$.

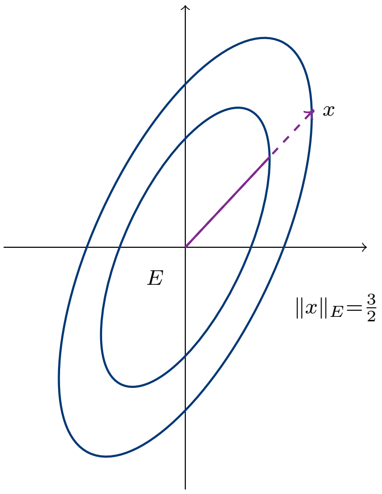

Figure 2.5.: **The Minkowski function of convex body.** If we scale the convex set E by the factor $3/2$, the vector x lies on its boundary.

Proof. Since $\mathcal{C}$ is a convex body, the Minkowski function $F_{\mathcal{C}}$ is well-defined. Particularly, since $\mathcal{C}$ is full-dimensional and closed, we have $F_{\mathcal{C}}(x) < \infty$ for all $x \in \mathbb{R}^n$. It remains to show that $F_{\mathcal{C}}$ satisfies the norm properties. The positivity and the absolute homogenity follow directly from the definition of $F_{\mathcal{C}}$. To prove the subadditivity, we consider two vectors $x, y \in \mathbb{R}^n$ with $F_{\mathcal{C}}(x) = \rho_1$ and $F_{\mathcal{C}}(y) = \rho_2$. Without loss of generality, we assume that $\rho_1, \rho_2 > 0$.
For all $\epsilon > 0$ it holds that $x \in (\rho_1 + \epsilon) \cdot \mathcal{C}$ and that $y \in (\rho_2 + \epsilon) \cdot \mathcal{C}$. Since $\rho_1, \rho_2 > 0$, it follows from the convexity of $\mathcal{C}$ that

$$x + y \in (\rho_1 + \rho_2 + 2\epsilon) \cdot \mathcal{C}.$$

Since $\epsilon > 0$ arbitrary, this shows that

$$F_{\mathcal{C}}(x + y) \leq \rho_1 + \rho_2 = F_{\mathcal{C}}(x) + F_{\mathcal{C}}(y).$$

□

Proposition 2.1.11 and Proposition 2.1.13 show the equivalence between norms and convex bodies symmetric about the origin.

2.1.3. Algorithmic aspects of norms and convex bodies

Since early all algorithms presented in this thesis involve norms or convex sets, we now consider some algorithmic aspects of norms and convex bodies.

If we consider computational statements, we always assume that all numbers we are dealing with are rationals. The *size of a rational number* $\alpha = p/q$ with $\gcd(p,q) = 1$ is defined as the maximum of the numerator and denominator in absolute values,

$$\text{size}(\alpha) := \max\{|p|, |q|\}.$$

By the *bit size* or the *representation size of a number* α, we mean $\log_2(\text{size}(\alpha))$. The size of a matrix or respectively a vector is the maximum of the size of its coordinates.

If we want to use norms in an algorithmic surrounding, we need some more requirements on the norm. First of all, we need to be able to compute the norm of a given vector efficiently. We call a norm which satisfies this property efficiently computable.

Definition 2.1.14. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. We call the norm* efficiently computable *if the norm function is polynomial time computable, i.e., if there exists an algorithm that given a vector $v \in \mathbb{Q}^n$ and an accuracy parameter δ outputs a number in the interval $[\|v\| \pm \delta]$ and the number of arithmetic operations of the algorithm is at most $(n \cdot \log_2(\text{size}(v)) \cdot \log_2(1/\delta))^{\mathcal{O}(1)}$.*

For the sake of simplicity, we will neglect the implementation detail of this definition in the following. We will assume that for an efficiently computable norm there exists an algorithm that given a vector $v \in \mathbb{Q}^n$ outputs $\|v\|$ and the number of arithmetic operations of the algorithm is $(n \cdot \log_2(\text{size}(v)))^{\mathcal{O}(1)}$.

Often, it is not sufficient that the norm function is efficiently computable. Additionally, we need some guarantees that the unit ball of the norm is in some way well-bounded. In this case we call a norm *tractable.* The following definition of a tractable norm is due to Goldreich and Goldwasser, see [GG00].

Definition 2.1.15. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. We call the norm* tractable *if it satisfies the following requirements:*

- *The norm function is efficiently computable.*
- *There exists a polynomial $c \in \mathbb{Z}[X]$ such that for all $x \in \mathbb{R}^n$*

$$2^{-c(n)}\|x\|_2 \leq \|x\| \leq 2^{c(n)}\|x\|_2.$$

We will later see that all standard norms, especially the so-called ℓ_p-norms are tractable norms.

If we require in the definition of a tractable norm that there exists a polynomial $c \in \mathbb{Z}[X]$ such that for all $x \in \mathbb{R}^n$ we have $2^{-c(n)}\|x\|_2 \leq \|x\| \leq 2^{c(n)}\|x\|_2$, then this is reflected in the fact that we often consider a family of norms. That means, we consider a sequence N_i, $i \in \mathbb{N}$ where $N_i : \mathbb{R}^i \to \mathbb{R}$ is a norm on $\mathbb{R}^i$. Of course, if we consider a fixed norm on $\mathbb{R}^n$ with n fixed, then c is always a constant.

The geometric interpretation of the second requirement of Definition 2.1.15 is that the unit ball $\bar{B}_n^{(\|\cdot\|)}(0,1)$ defined by the norm $\|\cdot\|$ contains a Euclidean ball with radius $2^{-c(n)}$ centered at the origin and is contained in a Euclidean ball with radius $2^{c(n)}$ centered at the origin,

$$\bar{B}_n^{(2)}(0, 2^{-c(n)}) \subseteq \bar{B}_n^{(\|\cdot\|)}(0,1) \subseteq \bar{B}_n^{(2)}(0, 2^{c(n)}).$$

In this case we call the ball $\bar{B}_n^{(\|\cdot\|)}(0,1)$ *well-bounded.*

Definition 2.1.16. *(Well-bounded convex set)*
A convex set $\mathcal{C}$ is called well-bounded *if the following information about $\mathcal{C}$ is given explicitly:*

- *an integer $n \in \mathbb{N}$ such that $\mathcal{C} \subseteq \mathbb{R}^n$,*
- *a positive rational number R such that $\mathcal{C} \subseteq \bar{B}_n^{(2)}(0,R)$, and*
- *a positive rational number r such that $\mathcal{C}$ contains a Euclidean ball with radius r. The center of this ball need not be known explicitly.*

In the following, whenever we say that a convex set $\mathcal{C}$ is well-bounded we mean that we know some parameter $n \in \mathbb{N}$ such that $\mathcal{C} \subseteq \mathbb{R}^n$, and that we are able to determine the numbers r and R explicitly from the shape of the convex body.

We will often assume that we are given the convex set in form of an oracle. That means, we have access to some algorithm which provides us some information about the convex set. For example the algorithm could be a "membership algorithm" that decides for a given vector whether it is contained in the convex set or not.
Mainly, we distinguish in the following between two different types of oracles.

Definition 2.1.17. *(Membership oracle of a convex set)*
A membership oracle of a convex set $\mathcal{C} \subseteq \mathbb{R}^n$ decides for a given vector $x \in \mathbb{R}^n$ whether x is contained in $\mathcal{C}$ or not.

Obviously, if a norm $\|\cdot\|$ on $\mathbb{R}^n$ is efficiently computable then we are able to realize an efficient membership oracle for the balls $B_n^{(\|\cdot\|)}(x,\alpha)$ with $x \in \mathbb{R}^n$ and $\alpha > 0$.

The second oracle also decides whether a given vector is contained in the convex set but it provides additionally some kind of a certificate if the vector is not contained in the convex set. This certificate is given in form of an affine hyperplane that separates this vector from the convex set.

Definition 2.1.18. *(Separation oracle of a convex set)*
A separation oracle of a convex set $\mathcal{C} \subseteq \mathbb{R}^n$ decides for a given vector $x \in \mathbb{R}^n$ whether x is contained in $\mathcal{C}$ or not. If $x \notin \mathcal{C}$, the oracle outputs an affine hyperplane that separates x from $\mathcal{C}$.

In general, we cannot realize an efficient separation oracle for balls $B_n^{(\|\cdot\|)}(x,\alpha)$ with $x \in \mathbb{R}^n$, $\alpha > 0$ for some efficiently computable norm $\|\cdot\|$ on $\mathbb{R}^n$. But one can show that for every efficiently computable norm we are able to realize an efficient separation oracle for the corresponding balls if we are able to compute efficiently a so-called subgradient of the norm, see [Lov86]. In Chapter 7, we will do this for generalization of ℓ_p-norms.

2.2. Special convex bodies and the corresponding norms

In the rest of this chapter, we consider some special classes of convex bodies and the corresponding norms that will be considered throughout this thesis. We start with the Euclidean norm and its generalization. The corresponding convex bodies of general Euclidean norms are ellipsoids.

2.2.1. Euclidean norms and ellipsoids

The Euclidean norm The most frequently used norm in $\mathbb{R}^n$ is the Euclidean norm $\|\cdot\|_2 : \mathbb{R}^n \to \mathbb{R}$, $x \mapsto \left(\sum_{i=1}^n |x_i|^2\right)^{1/2}$. The corresponding unit ball of the Euclidean norm is the Euclidean unit ball

$$\bar{B}_n^{(2)}(0,1) = \{x \in \mathbb{R}^n | \|x\|_2 \leq 1\} = \{x \in \mathbb{R}^n | x^T x \leq 1\}.$$

The surface of this ball is denoted by $\mathbb{S}^{n-1} := \mathbb{S}^{n-1}(1)$. We define

$$\mathbb{S}^{n-1}(\alpha) := \{x \in \mathbb{R}^n |\ \|x\|_2 = \alpha\}.$$

The volume of the Euclidean unit ball is

$$\mathrm{vol}_n\left(B_n^{(2)}(0,1)\right) = \frac{\pi^{n/2}}{\Gamma\left(1+\frac{n}{2}\right)},$$

where $\Gamma(\cdot)$ denotes the Gamma function, see Section A.0.3 in the Appendix.

The important property of the Euclidean norm is that it is based on an inner product. An *inner product* on $\mathbb{R}^n \times \mathbb{R}^n$ is a symmetric bilinear mapping $s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ which satisfies the following properties:

- $s(x,x) \geq 0$ for all $x \in \mathbb{R}^n$ and $s(x,x) = 0$ if and only if $x = 0$,
- $s(\theta \cdot x, y) = \theta \cdot s(x,y)$ for all $x, y \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$,
- $s(x+y, z) = s(x,z) + s(y,z)$ for all $x, y, z \in \mathbb{R}^n$, and
- $s(x,y) = s(y,x)$ for all $x, y \in \mathbb{R}^n$.

Based on an inner product $s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ we can define a norm on $\mathbb{R}^n$ by

$$\mathbb{R}^n \to \mathbb{R},\ x \mapsto \sqrt{s(x,x)}.$$

Definition 2.2.1. *(Norm induced by an inner product)*
A norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ *is induced by an inner product if there exists an inner product* $s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *such that* $\|x\| = \sqrt{s(x,x)}$ *for all* $x \in \mathbb{R}^n$.

For the Euclidean norm, we have $\|x\|_2 = \sqrt{\langle x, x\rangle}$ for all $x \in \mathbb{R}^n$, where $\langle \cdot, \cdot \rangle$ is defined as the scalar product

$$\langle x, y\rangle := \sum_{i=1}^{n} x_i \cdot y_i = x^T y$$

for $x, y \in \mathbb{R}^n$. In the following, we will use both representations of the scalar product, $\langle x, y\rangle$ as well as $x^T y$, depending on what is more suitable in the context.

Lemma 2.2.2. *(Cauchy-Schwarz-inequality)*
For $x, y \in \mathbb{R}^n$ *we have*

$$|\langle x, y\rangle| \leq \|x\|_2 \cdot \|y\|_2.$$

The Euclidean norm on $\mathbb{R}^n$ induces a matrix norm on $\mathbb{R}^{n\times n}$, i.e., a norm $\mathbb{R}^{n\times n} \to \mathbb{R}$. For a matrix $A \in \mathbb{R}^{n\times n}$, we set

$$\|A\|_2 := \max\left\{ \frac{\|Ax\|_2}{\|x\|_2} \;\middle|\; x \in \mathbb{R}^n \backslash \{0\} \right\}.$$

Obviously, this mapping defines a norm on $\mathbb{R}^{n\times n}$, see for example [SK09]. The matrix norm $\|\cdot\|_2$ induced by the Euclidean norm is called the *spectral norm* of the matrix A. The Euclidean norm and the spectral norm are *compatible*, that means we have

$$\|Ax\|_2 \leq \|A\|_2 \cdot \|x\|_2$$

for all $A \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$. Using symmetric positive definite matrices, we can develop an alternative characterization of the spectral norm of a matrix.

Definition 2.2.3. *(Symmetric positive definite matrices)*
A matrix $D \in \mathbb{R}^{n\times n}$ *is called* symmetric positive definite *if it is symmetric and if* $x^T D x > 0$ *for all* $x \in \mathbb{R}^n \backslash \{0\}$.

Every symmetric positive definite matrix is nonsingular and the inverse matrix is also symmetric positive definite. Furthermore, all eigenvalues of a symmetric positive definite matrix are positive real numbers and for every symmetric positive definite matrix $D \in \mathbb{R}^{n\times n}$ there exists a decomposition $D = Q^T \cdot Q$, where $Q \in \mathbb{R}^{n\times n}$. One can show that there exists a uniquely determined symmetric positive definite matrix X such that $D = X^T \cdot X = X \cdot X$. We call X the *square root* of D, denoted by $D^{1/2}$, see [HJ85].

Based on the observation that the Euclidean norm is induced by an inner product we can show that spectral norm $\|A\|_2$ of a matrix A is the square root of the largest eigenvalue of the symmetric positive definite matrix $A^T A$. For $A \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$, we have $\|Ax\|_2 = \sqrt{x^T A^T A x}$ and the matrix $A^T A$ is symmetric positive definite.

Lemma 2.2.4. *Let $D \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Then we have*

$$\max\left\{\frac{x^T Dx}{x^T x} \;\middle|\; x \in \mathbb{R}^n \backslash \{0\}\right\} = \sqrt{\eta_n(D)},$$

where $\eta_n(D)$ is the largest eigenvalue of the matrix D.

Proof. Since the matrix D is symmetric positive definite, there exists an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that

$$D = Q \cdot \Lambda \cdot Q^T$$

where $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix which consists of the real positive eigenvalues $\eta_1(D), \dots, \eta_n(D)$ of the matrix D. This result is known as the spectral theorem, see e. g. [Str06]. Thus, for every vector $x \in \mathbb{R}^n$ we obtain that

$$x^T Dx = x^T Q \cdot \Lambda \cdot Q^T x = (Q^T x)\Lambda(Q^T x).$$

If we set $y := Q^T x \in \mathbb{R}^n$, we have

$$x^T Dx = \sum_{i=1}^{n} \eta_i(D) \cdot y_i^2.$$

Without loss of generality, we assume that $\eta_n(D)$ is the largest eigenvalue of the matrix D,

$$x^T Dx = \sum_{i=1}^{n} \eta_i(D) \cdot y_i^2 \leq \eta_n(D) \sum_{i=1}^{n} y_i^2.$$

By definition, we have

$$\sum_{i=1}^{n} y_i^2 = y^T \cdot y = x^T Q Q^T x = x^T x$$

and it follows that

$$x^T Dx \leq \eta_n(D) \cdot x^T x.$$

This shows that for all $x \in \mathbb{R}^n \backslash \{0\}$ we have

$$\frac{x^T Dx}{x^T x} \leq \eta_n(D).$$

Furthermore, the vector $x = Q \cdot e_n \in \mathbb{R}^n$ satisfies

$$\frac{x^T Dx}{x^T x} = \frac{e_n^T Q^T D Q e_n}{e_n^T Q^T Q e_n} = e_n^T \Lambda e_n = \eta_n(D),$$

which shows that

$$\max\left\{\sqrt{\frac{x^T Dx}{x^T x}} \;\middle|\; x \in \mathbb{R}^n \backslash \{0\}\right\} = \sqrt{\eta_n(D)}.$$

□

Using this result with $D = A^T \cdot A$, it follows that the spectral norm of the matrix A is the square root of the largest eigenvalue of the matrix $A^T A$,

$$\|A\|_2 = \eta_n(A^T A).$$

If A is symmetric, we obtain $\|A\|_2 = \eta_n(A)$. If A is nonsingular, the spectral norm of the inverse matrix A^{-1} is the square root of the largest eigenvalue of $(A^T)^{-1}A^{-1} = (AA^T)^{-1}$. Since the eigenvalues of $(AA^T)^{-1}$ are the inverse of the eigenvalues of the matrix AA^T, the spectral norm of A^{-1} is the inverse of the smallest eigenvalue of the matrix AA^T, $\|A^{-1}\|_2 = (\eta_1(AA^T))^{-1}$. The matrices $A^T A$ and AA^T are similar since $A^{-1}(AA^T)A = A^T A$, which means that they have the same eigenvalues. From this, it follows that

$$\|A^{-1}\|_2 = \frac{1}{\sqrt{\eta_1(A^T A)}},$$

where $\eta_1(A^T A)$ is the smallest eigenvalue of $A^T A$. Since $A^T A$ is symmetric positive definite, we have $\eta_1(A^T A) > 0$.

General Euclidean norms Using symmetric positive definite matrices, we can define generalizations of the Euclidean unit ball. The corresponding norms are also based on an inner product.

For a symmetric positive definite matrix $D \in \mathbb{R}^{n\times n}$ we define the mapping

$$\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R},\ (x, y) \mapsto x^T D y.$$

Since D is positive definite, we have $x^T Dx > 0$ for all $x \in \mathbb{R}^n\backslash\{0\}$ and $x^T Dx = 0$ for $x = 0$. Obviously, this mapping satisfies also the other required properties of an inner product. This shows that the mapping

$$\|\cdot\|_D : \mathbb{R}^n \to \mathbb{R},\ x \mapsto \sqrt{x^T D^{-1} x}$$

defines a norm on $\mathbb{R}^n$. Norms of this type are called *general Euclidean norms.* At a first glance, it is not clear why we use here the matrix D^{-1} in the definition instead of the matrix D. But later, we will see that many properties of this norm and the corresponding convex body can be deduced directly from the properties of the matrix D.

For general Euclidean norms we can generalize the Cauchy-Schwarz inequality presented in Lemma 2.2.2.

Lemma 2.2.5. *(Generalized Cauchy-Schwarz inequality)*
Let $D \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. For all $x, y \in \mathbb{R}^n$ we have that

$$|x^T y| \leq \sqrt{x^T D^{-1} x} \cdot \sqrt{y^T D y}.$$

Proof. Since A is symmetric positive definite, there exists a uniquely determined square root $D^{1/2}$. Using the Cauchy-Schwarz inequality, see Lemma 2.2.2, we obtain

$$\begin{aligned}|x^Ty| &= |x^TD^{-1/2}D^{1/2}y^T| \\ &= |\langle D^{-1/2}x, D^{1/2}y\rangle| \\ &\leq \|D^{-1/2}x\|_2 \cdot \|D^{1/2}y\|_2.\end{aligned}$$

By definition of the Euclidean norm, the statement follows

$$\begin{aligned}|x^Ty| &\leq \sqrt{(D^{-1/2}x)^T(D^{-1/2}x)} \cdot \sqrt{(D^{1/2}y)^T(D^{1/2}y)} \\ &\leq \sqrt{x^TD^{-1}x} \cdot \sqrt{y^TD^{-1}y}.\end{aligned}$$

□

Sometimes general Euclidean norms are also called *ellipsoidal norms* due to the fact that the corresponding unit balls of an ellipsoidal norm are ellipsoids centered at the origin.

Definition 2.2.6. *(Ellipsoids)*
A set $E \subseteq \mathbb{R}^n$ is called an ellipsoid if there exists a vector $c \in \mathbb{R}^n$ and a symmetric positive definite matrix $D \in \mathbb{R}^{n\times n}$ such that

$$E = \big\{x \in \mathbb{R}^n |\ (x-c)^TD^{-1}(x-c) \leq 1\big\}.$$

The vector c is called the center of the ellipsoid and we denote by $E(D,c)$ the ellipsoid given by the matrix D and the vector c.

The decomposition of a symmetric positive definite matrix $D = Q^T \cdot Q$ can be used to define a bijective affine transformation that maps the Euclidean unit ball to the ellipsoid $E(D,c)$, see Figure 2.6 for an illustration. This leads to an alternative characterization of an ellipsoid.

Lemma 2.2.7. *A set $E \subseteq \mathbb{R}^n$ is an ellipsoid $E = E(D,c)$ for a symmetric positive definite matrix $D \in \mathbb{R}^{n\times n}$ and a vector $c \in \mathbb{R}^n$ if and only if E is the affine image of the Euclidean unit ball, i.e.,*

$$E = Q^T \cdot \bar{B}_n^{(2)}(0,1) + c = Q^T\bar{B}_n^{(2)}((Q^T)^{-1}c, 1),$$

where $D = Q^T \cdot Q$.

Proof. We consider a matrix Q such that $D = Q^T \cdot Q$. A vector $x \in \mathbb{R}^n$ is contained in the ellipsoid $E = E(D,c)$ if and only if $(x-c)^TD^{-1}(x-c) \leq 1$. By straightforward calculation, we see that

$$\begin{aligned}(x-c)^TD^{-1}(x-c) &= (x-c)^TQ^{-1}(Q^T)^{-1}(x-c) \\ &= \left((Q^T)^{-1}(x-c)\right)^T\left((Q^T)^{-1}(x-c)\right).\end{aligned}$$

This shows that the vector $x \in \mathbb{R}^n$ is contained in $E(D,c)$ if and only if $\|(Q^T)^{-1}(x-c)\|_2 \leq 1$ or equivalently if x is of the form $x = Q^Ty + c$, where $y \in \bar{B}_n^{(2)}(0,1)$. □

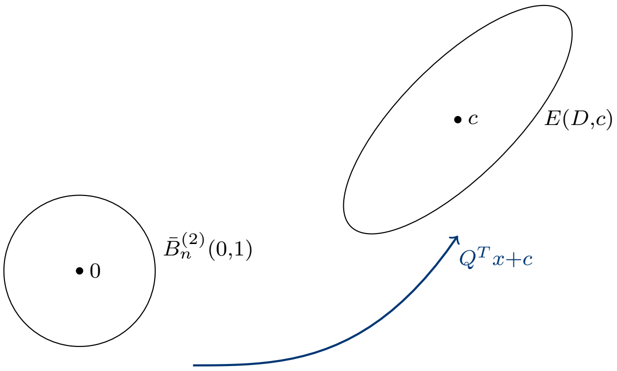

Figure 2.6.: **Characterization of an ellipsoid.** The ellipsoid $E(D,c)$ is the image of the Euclidean unit ball $\bar{B}_n^{(2)}(0,1)$ under the affine bijective transformation $x \mapsto Q^T x + c$, where $D = Q^T \cdot Q$.

This relation between ellipsoids and the Euclidean unit ball is fundamental in the understanding of ellipsoids. Nearly every property of an ellipsoid can be deduced from the corresponding property of the Euclidean unit ball by applying the transformation Q^T. For example, we are able to compute the volume of an ellipsoid.

Lemma 2.2.8. *Let $E(D,c) \subseteq \mathbb{R}^n$ be an ellipsoid. The volume of this ellipsoid is*

$$\mathrm{vol}_n(E(D,c)) = \sqrt{\det(D)} \cdot \mathrm{vol}_n(B_n^{(2)}(0,1)).$$

Proof. Let $D = Q^T \cdot Q$ be an arbitrary decomposition of the matrix D defining the ellipsoid. Then the ellipsoid $E(D,c)$ is the image of the Euclidean unit ball under the bijective affine transformation $x \mapsto Q^T x + c$, see Lemma 2.2.7. Hence, the statement follows directly from Lemma 2.1.3 using that

$$|\det(Q^T)| = \sqrt{\det(Q^T)^2} = \sqrt{\det(Q) \cdot \det(Q^T)} = \sqrt{\det(D)}.$$

□

Furthermore, it follows that the relation between the volumes of two ellipsoids $E_1, E_2 \subseteq \mathbb{R}^n$ is invariant under affine bijective transformation, i.e.,

$$\frac{\mathrm{vol}_n(E_1)}{\mathrm{vol}_n(E_2)} = \frac{\mathrm{vol}_n(T(E_1))}{\mathrm{vol}_n(T(E_2))}, \tag{2.2}$$

where $T : \mathbb{R}^n \to \mathbb{R}^n$ is a bijective affine transformation.

Since the decomposition of a symmetric positive definite matrix D in $D = Q^T \cdot Q$ is not uniquely determined, there exists many transformations that map the Euclidean unit ball to the same ellipsoid $E(D,c)$. Each decomposition $D = Q^T \cdot Q$ of the matrix D

defines a different transformation. However, the decomposition of a symmetric positive definite matrix is unique except for multiplication with an orthogonal matrix O, i.e., a matrix satisfying $O^T \cdot O = I_n$, where I_n is the identity matrix in $\mathbb{R}^n$. Geometrically, this multiplication with an orthogonal matrix rotates the Euclidean unit ball before we apply the transformation. That means, the bijective affine transformation $\bar{B}_n^{(2)}(0,1) \to E(D,c)$ is unique up to rotation of the Euclidean unit ball.

In the following, we will sometimes need a bijective affine transformation $\tau : \bar{B}_n^{(2)}(0,1) \to E(D,c)$ satisfying an additional property: Given two vectors $e \in \bar{B}_n^{(2)}(0,1)$ and $d \in E(D,c)$ with $e^T \cdot e = (d-c)^T D^{-1}(d-c)$ we are searching for a transformation τ such that $\tau(e) = d$. Such a transformation can be found as follows: We start with an arbitrary bijective affine transformation

$$\tau : \bar{B}_n^{(2)}(0,1) \to E(D,c),\ x \mapsto Q^T x + c$$

and compute the preimage of the vector d under this transformation,

$$\tau^{-1}(d) = (Q^T)^{-1}(d-c) \in \bar{B}_n^{(2)}(0,1).$$

Now we compute an orthogonal matrix $O \in \mathbb{R}^{n\times n}$ such that

$$(Q^T)^{-1}(d-c) = O \cdot e.$$

Such an orthogonal matrix exists since the vectors $(Q^T)^{-1}(d-c)$ and e have the same length,

$$\begin{aligned} \|(Q^T)^{-1}(d-c)\|_2^2 &= \langle (Q^T)^{-1}(d-c), (Q^T)^{-1}(d-c)\rangle \\ &= (d-c)^T Q^{-1}(Q^T)^{-1}(d-c) \\ &= (d-c)^T D^{-1}(d-c) \\ &= e^T \cdot e = \|e\|_2^2. \end{aligned}$$

Then, the bijective affine transformation

$$\bar{\tau} : x \mapsto (O^T \cdot Q)^T x + c = Q^T O x + c$$

maps the vector $e \in \bar{B}_n^{(2)}(0,1)$ to the vector $d \in E(D,c)$,

$$Q^T O e + c = Q^T (Q^T)^{-1}(d-c) + c = d.$$

This proves the following statement.

Lemma 2.2.9. *Let $E(D,c) \subseteq \mathbb{R}^n$ an ellipsoid given by $D \in \mathbb{R}^{n\times n}$ symmetric positive definite and $c \in \mathbb{R}^n$. Let $e, d \in \mathbb{R}^n$ satisfying*

$$e^T e = (d-c)^T D^{-1}(d-c) \le 1.$$

Then there exists a bijective affine transformation $\bar{\tau} : \bar{B}_n^{(2)}(0,1) \to E(D,c)$ satisfying $\bar{\tau}(e) = d$.

Obviously, every norm defined by an ellipsoid $E(D, 0)$ is efficiently computable. Furthermore, we can explicitly determine the radius of a circumscribed and of an inscribed Euclidean ball. The radius of the circumscribed Euclidean ball is given by the square root of the largest eigenvalue of the matrix D defining the ellipsoid, whereas the radius of the inscribed Euclidean ball is the square root of the smallest eigenvalue of D.

Lemma 2.2.10. *Let $E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid given by $D \in \mathbb{R}^{n\times n}$ symmetric positive definite and $c \in \mathbb{R}^n$. Then*

$$\bar{B}_n^{(2)}\big(c, \sqrt{\eta_1(D)}\big) \subseteq E(D, c) \subseteq \bar{B}_n^{(2)}\big(c, \sqrt{\eta_n(D)}\big),$$

where $\eta_1(D)$ denotes the smallest and $\eta_n(D)$ the largest eigenvalue of D.

Since D is symmetric positive definite all eigenvalues of D are real and positive, i.e., the radii of the Euclidean balls are well-defined. As we have seen before, $\eta_n(D)$ is the spectral norm of the matrix D and $\eta_1(D)$ is the spectral norm of the inverse matrix D^{-1}. Hence, Lemma 2.2.10 shows that the ellipsoid $E(D, c)$ is contained in a Euclidean ball with radius $\|D\|_2$ and contains a Euclidean ball with radius $\|D^{-1}\|_2$.

Proof. Without loss of generality, we assume that $c = 0$. Since $D \in \mathbb{R}^{n\times n}$ is symmetric positive definite, for all $x \in \mathbb{R}^n\backslash\{0\}$ we have that

$$x^T D^{-1} x \leq \eta_n(D^{-1}) \cdot x^T x$$

as we have already seen in Lemma 2.2.4. Since the largest eigenvalue of the matrix D^{-1} is the inverse of the smallest eigenvalue of the matrix D, $\eta_n(D^{-1}) = 1/\eta_1(D)$, we have

$$x^T D^{-1} x \leq \frac{1}{\eta_1(D)} x^T x \text{ for all } x \in \mathbb{R}^n\backslash\{0\}.$$

This shows that every vector $x \in \bar{B}_n^{(2)}(0, \sqrt{\eta_1(D)})$ satisfies $x^T D^{-1} x \leq 1$, since $x^T x \leq \eta_1(D)$. This shows that $x \in E(D, 0)$.
With the same argumentation as in the proof of Lemma 2.2.4 we can show that

$$x^T D^{-1} x \geq \eta_1(D^{-1}) x^T x$$

for all $x \in \mathbb{R}^n$. Again using that $\eta_1(D^{-1}) = 1/\eta_n(D)$, we obtain that

$$x^T D^{-1} x \geq \frac{1}{\eta_n(D)} x^T x \text{ for all } x \in \mathbb{R}^n\backslash\{0\},$$

from which it follows that $x^T x \leq \eta_n(D)$ for $x \in E(D, 0)$. □

A fundamental observation is that every full-dimensional bounded convex set can be approximated using an ellipsoid. By the approximation of a bounded convex set $\mathcal{C}$ by an ellipsoid E we understand an ellipsoid which is contained in the convex set, $E \subseteq \mathcal{C}$. The approximation factor is that factor, which we need to scale the ellipsoid with such that the scaled ellipsoid contains the convex set.

By scaling an ellipsoid with a positive factor $r > 0$ we understand the ellipsoid obtained from E by scaling it from its center by the factor r. We denote this as $r \star E$. Formally, if $E = E(D, c)$, then

$$r \star E := r \cdot E(D, 0) + c.$$

Lemma 2.2.11. *Let $E = E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid and $r > 0$. Then,*

$$r \star E = E(r^2 \cdot D, c).$$

Proof. Without loss of generality we assume that $c = 0$. A vector $x \in \mathbb{R}^n$ is contained in the ellipsoid $r \star E$ if and only if $x \in r \cdot E(D, 0)$, i.e., if there exists a vector $y \in E(D, 0)$ such that $x = r \cdot y$. By definition of an ellipsoid, the vector y satisfies $y^T D^{-1} y \leq 1$. Using the following rearrangements

$$y^T D^{-1} y = \left(\frac{x}{r}\right)^T D^{-1} \left(\frac{x}{r}\right) = x^T \cdot \left(\frac{1}{r^2} D^{-1}\right) \cdot x = x^T \cdot \left(r^2 D\right)^{-1} x,$$

we get that the vector x is contained in $r \star E$ if and only if $x^T (r^2 D)^{-1} x \leq 1$. □

An ellipsoid which approximates a bounded convex set is called an approximate Löwner-John ellipsoid.

Definition 2.2.12. *(Approximate Löwner-John ellipsoid)*
Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional bounded convex set and $0 < \gamma < 1$. An ellipsoid E satisfying $E \subseteq \mathcal{C} \subseteq (1/\gamma) \star E$ is called $1/\gamma$-approximate Löwner-John ellipsoid of $\mathcal{C}$. We call $1/\gamma$ the approximation factor (of the Löwner-John ellipsoid).

If we consider a γ-approximate Löwner-John ellipsoid of a convex set $\mathcal{C}$ where the approximation factor γ is optimal, we call this ellipsoid the Löwner-John ellipsoid. Here, optimal means that for all $\gamma' < \gamma$ there does not exist an ellipsoid E' satisfying $E' \subseteq \mathcal{C} \subseteq (1/\gamma') \star E'$.
A fundamental result is that every full-dimensional bounded convex set in $\mathbb{R}^n$ can be approximated by an ellipsoid with approximation factor of at most n. For symmetric convex sets, the approximation factor can be improved to $\sqrt{n}$. Results of this type have been proved independently by several persons, see [DGK63]. The first result of this type is attributed to Löwner. The following theorem is due to John, see [Joh48]. A proof of it can be found in [Bal97].

Theorem 2.2.13. *(John's Lemma)*
Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional closed, bounded convex set. Then there exists an ellipsoid $E \subseteq \mathbb{R}^n$ such that

$$E \subseteq \mathcal{C} \subseteq n \star E.$$

If $\mathcal{C}$ is symmetric about the origin, the approximation factor can be improved by the factor $\sqrt{n}$, i.e., in this case there exists an ellipsoid E satisfying

$$E \subseteq \mathcal{C} \subseteq \sqrt{n} \star E.$$

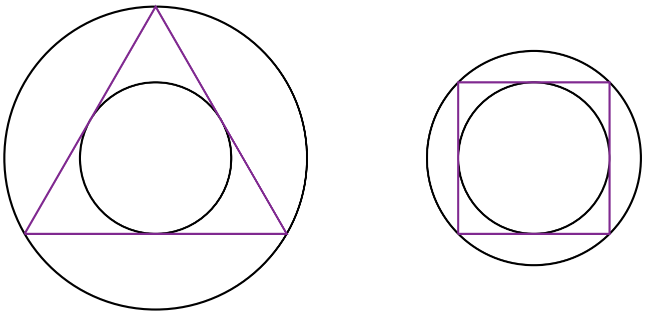

Figure 2.7.: **John's lemma.** On the left, we see that for a regular simplex in $\mathbb{R}^2$ the ratio between a inscribed and circumscribed circle is exactly 2 and that it is the best possible. On the right, we consider a symmetric cube where the ratio between the inscribed and circumscribed circle is exactly $\sqrt{2}$.

In general, these approximation factors are best possible. An example for the optimality are the regular simplex or the cube, see Figure 2.7.

A Löwner-John ellipsoid of a bounded convex set can also be characterized as follows: The inscribed ellipsoid E is the ellipsoid contained in E with maximal volume, also called maximum volume ellipsoid. Analogously, the circumscribed ellipsoid $\gamma \star E$ is the ellipsoid with minimal volume that contains E, also called the minimum volume ellipsoid. For more details about John's lemma and its consequences, see [Mat02], [Bar02] or [Bal97].

Unfortunately, the proof of John's lemma is not constructive. In general, the computation of an approximate Löwner-John ellipsoid is hard. For example, for a given finite set of points it is NP-hard to compute the smallest enclosing ellipsoid if the dimension is part of the input, see [Mat02].

We observe that if we are given a γ-approximate Löwner-John ellipsoid for a full-dimensional bounded convex set, we are able to compute an inscribed and circumscribed Euclidean ball for the convex set, i.e., the convex set is well-bounded.

Lemma 2.2.14. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional bounded convex set and $E(D,c) \subseteq \mathbb{R}^n$ be a γ-approximate Löwner-John ellipsoid of $\mathcal{C}$ for some parameter $\gamma \geq 1$. Then*

$$\bar{B}_n^{(2)}(c, \sqrt{\eta_1(D)}) \subseteq \mathcal{C} \subseteq \bar{B}_n^{(2)}(c, \gamma \cdot \sqrt{\eta_n(D)}),$$

where $\eta_1(D)$ is the smallest and $\eta_n(D)$ is the largest eigenvalue of the matrix D.

The proof of this lemma follows directly from Lemma 2.2.10 together with the fact that $E(D,c) \subseteq \mathcal{C} \subseteq E(\gamma^2 \cdot D, c)$.

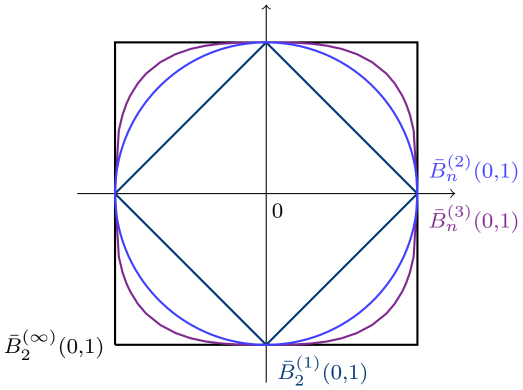

Figure 2.8.: **Unit balls of different norms.** The picture shows the unit ball of the ℓ_1-norm, the Euclidean norm, the ℓ_3-norm, and the ℓ_∞-norm.

2.2.2. ℓ_p-norms and ℓ_p-balls with $1 \le p \le \infty$

The mostly used non-Euclidean norms are arbitrary ℓ_p-norms with $1 \le p \le \infty$. The ℓ_p-norm of a vector $x \in \mathbb{R}^n$ is defined by

$$\|x\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

for $1 \le p < \infty$ and

$$\|x\|_\infty := \max\{|x_i||1 \le i \le n\}.$$

For $p = 2$ we obtain the Euclidean norm. The balls generated by an ℓ_p-norm with $1 \le p \le \infty$ are called ℓ_p-balls. We denote them by $B_n^{(p)}(x, \alpha)$ or $\bar{B}_n^{(p)}(x, \alpha)$ respectively. The unit balls of some ℓ_p-norms are illustrated in Figure 2.8.

It follows immediately from its definition that the function $\| \cdot \|_p$ satisfies the first two properties of a norm, positivity and absolute homogenity. The proof of the third property, the subadditivity, is more substantial. It is based on Hölder's inequality.

Proposition 2.2.15. *(Hölder's inequality)*
Let $1 \le p, q \le \infty$ with $1/p + 1/q = 1$. We set $1/\infty = 0$. For all $x, y \in \mathbb{R}^n \backslash \{0\}$ we have

$$|\langle x, y\rangle| \le \|x\|_p \cdot \|y\|_q.$$

The inequality is fulfilled with equality if and only if there exists $\theta \in \mathbb{R}$ such that

$$\theta \cdot x_k^{1/p} = y_k^{1/q} \textit{ for all } 1 \le k \le n.$$

Originally, this inequality was obtained by Rogers in 1888. One year later, in 1889, it was derived in another way by Hölder. The form as it is presented above is due to Riesz, who also recognized its fundamental role. Thus, the inequality might be better called Rogers-Hölder-Riesz inequality. For a proof of the inequality see for example [Ste04]. Based on Hölder's inequality we can prove the following statement.

Proposition 2.2.16. *(Minkowski's Inequality)*
Let $x, y \in \mathbb{R}^n$ and $p \geq 1$. Then it holds that

$$\|x + y\|_p \leq \|x\|_p + \|y\|_p.$$

Moreover, if $p > 1$ and $x \neq 0$, the inequality is fulfilled with equality if and only if there exists a constant $\theta \in \mathbb{R}$ such that $|x_k| = |\theta| \cdot |y_k|$ for all $1 \leq k \leq n$ and x_k and y_k have the same sign for each $1 \leq k \leq n$.

The proof of Minkowski's inequality follows from Hölder's inequality and can also be found in [Ste04]. In particular, Minkowski's inequality shows that the ℓ_p-norms with $1 < p < \infty$ are strictly convex, i.e., for $x, y \in \mathbb{R}^n \backslash \{0\}$, $x \neq y$ it holds that

$$\|x + y\|_p < \|x\|_p + \|y\|_p$$

or equivalently that

$$\|\theta x + (1 - \theta)y\|_p < \theta\|x\|_p + (1 - \theta)\|y\|_p$$

for all $0 < \theta < 1$. In contrast, the ℓ_1-norm and the ℓ_∞-norm are not strictly convex. Geometrically, we see this since the boundaries of their unit balls contain straight lines. Analytically, we have for the ℓ_∞-norm that

$$\|\frac{1}{2}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix}\right)\|_\infty = 1 = \frac{1}{2}\|\begin{pmatrix} 1 \\ 1 \end{pmatrix}\|_\infty + \frac{1}{2}\|\begin{pmatrix} -1 \\ 1 \end{pmatrix}\|_\infty$$

respectively. For the ℓ_1-norm, we obtain a similar result if we consider the vectors $(1, 0)^T$ and $(0, 1)^T$.

We observe that for $p < 1$, the function $\mathbb{R}^n \to \mathbb{R}$, $x \mapsto (\sum_{i=1}^n |x_i|^p)^{1/p}$ does not define a norm since this function is not convex and does not satisfy the triangle inequality.

All ℓ_p-norms with $1 \leq p \leq \infty$ are efficiently computable. Furthermore, we obtain as a special case of Hölder's inequality a relation between the Euclidean norm and arbitrary ℓ_p-norms: For all $x \in \mathbb{R}^n$ we have

$$\|x\|_2 \leq \|x\|_p \leq n^{1/p - 1/2}\|x\|_2$$

if $1 \leq p \leq 2$ and

$$n^{1/p - 1/2}\|x\|_2 \leq \|x\|_p \leq \|x\|_2$$

if $2 < p < \infty$. For the ℓ_∞-norm it holds that

$$n^{-1/2}\|x\|_2 \leq \|x\|_\infty \leq \|x\|_2.$$

This shows that all ℓ_p-norms are tractable norms and that the corresponding unit balls are well-bounded.

We have already observed that the ℓ_1-norm and the ℓ_∞-norm are in some kind special ℓ_p-norms since they are not strictly convex. Furthermore, their corresponding unit balls are polytopes.

2.2.3. Polyhedral norms and polytopes

A *polyhedron* is the solution set of a system of inequalities given by a matrix $A \in \mathbb{R}^{s\times n}$ and a vector $\beta \in \mathbb{R}^s$,

$$\{x \in \mathbb{R}^n | A \cdot x \leq \beta\}. \tag{2.3}$$

Obviously, a set $P \subseteq \mathbb{R}^n$ is a polyhedron if and only if it can be represented as the intersection of finitely many halfspaces. A bounded polyhedron is called a *polytope*.

Mainly there are two possible ways how a polyhedron can be described: As the solution set of a system of inequalities as it is done in (2.3) or as the convex hull of finitely many vectors. This is illustrated by the unit ball of the ℓ_1-norm and the unit ball of the ℓ_∞-norm.

- The polytope which defines the ℓ_1-norm is given by the 2^n constraints $\langle x, e\rangle \leq 1$, where $e \in \{\pm 1\}^n$, i.e.,

  $$\bar{B}_n^{(1)}(0,1) = \{x \in \mathbb{R}^n | \langle x, e\rangle \leq 1 \text{ for all } e \in \{\pm 1\}^n\}. \tag{2.4}$$

  Alternatively, $\bar{B}_n^{(1)}(0,1)$ can be described as the convex hull of the $2n$ unit vectors $\pm e_i \in \mathbb{R}^n$ where $1 \leq i \leq n$, i.e., $\bar{B}_n^{(1)}(0,1) = \text{conv}(\{\pm e_i | 1 \leq i \leq n\})$.

- In contrast, the polytope which defines the ℓ_∞-norm is given by the $2n$ constraints $\langle x, e_i\rangle \leq 1$ and $\langle x, -e_i\rangle \leq 1$ for $1 \leq i \leq n$,

  $$\bar{B}_n^{(\infty)}(0,1) = \{x \in \mathbb{R}^n | \langle x, e_i\rangle \leq 1 \text{ and } \langle x, -e_i\rangle \leq 1 \text{ for all } 1 \leq i \leq n\}. \tag{2.5}$$

  Accordingly, $\bar{B}_n^{(\infty)}(0,1)$ can be described as the convex hull of the 2^n vectors $\{\pm 1\}^n$, $\bar{B}_n^{(\infty)}(0,1) = \text{conv}(\{\pm 1\}^n\})$.

In this thesis, we will always assume that a polyhedron is given in the first way, i.e., in the form as it is described in (2.3). Thus, whenever we speak in the following about the polytope which defines the ℓ_1-norm or the ℓ_∞-norm, we always assume that they are given as in (2.4) or (2.5) respectively.

Since we are interested in computational statements, we always assume that the polyhedron is given by a rational matrix $A \in \mathbb{Q}^{s \times n}$ and a rational vector $\beta \in \mathbb{Q}^n$. We denote by the size of a polyhedron P the maximum of n, s, and the size of the coordinates of A and β, i.e., if $P = \{x \in \mathbb{R}^n | A \cdot x \leq \beta\}$ with $A \in \mathbb{Q}^{s \times n}$ and $\beta \in \mathbb{Q}^n$ then $\text{size}(P) := \max\{n, s, \text{size}(A), \text{size}(\beta)\}$.

Given a full-dimensional polytope P symmetric about the origin, we call the corresponding norm a *polyhedral norm*, denoted by $\| \cdot \|_P$. Particularly, the ℓ_1-norm and the ℓ_∞-norm are polyhedral norms, whereas ℓ_p-norms with $1 < p < \infty$ are not polyhedral norms.

In the rest of this section, we show that polytopes symmetric about the origin are well-bounded convex bodies, i.e., that we compute the radius of a inscribed and of a circumscribed Euclidean ball.
If P is a polytope symmetric about the origin, there exists a parameter $s \in \mathbb{N}$ and a set of constraints $H_P = \{h_1, \ldots, h_{s/2}\} \subseteq \mathbb{R}^n$ such that P is the intersection of a collection of halfspaces determined by H_P,

$$\begin{aligned} P &= \bigcap_{i=1}^{s/2} \{x \in \mathbb{R}^n | \langle x, h_i \rangle \leq 1\} \cap \bigcap_{i=1}^{s/2} \{x \in \mathbb{R}^n | \langle x, h_i \rangle \geq -1\} \\ &= \{x \in \mathbb{R}^n | \langle x, h_i \rangle \leq 1 \text{ and } \langle x, -h_i \rangle \leq 1 \text{ for } 1 \leq i \leq s/2\}, \end{aligned}$$

see [BJWW98]. Here, s is the number of facets of the polytope, i.e., the number of $(n-1)$-dimensional faces of the polytope. A face of a polytope P is a set $F \neq \emptyset$ satisfying $F = \{x \in P | \langle x, h_i \rangle = 1 \text{ or } \langle x, h_i \rangle = -1 \text{ for some } 1 \leq i \leq s/2\}$.
If we deal with algorithms, we can always assume that $H_P \subseteq \mathbb{Q}^n$. Sometimes, it will be easier to assume that $H_P \subseteq \mathbb{Z}^n$. Then, the polyhedron P is given by a set $H_P \subseteq \mathbb{Z}^n$ together with a set of parameters $\{\beta_1, \ldots, \beta_{s/2}\} \subseteq \mathbb{N}$,

$$P = \{x \in \mathbb{R}^n | \langle x, h_i \rangle \leq \beta_i \text{ and } \langle x, -h_i \rangle \leq \beta_i \text{ for all } 1 \leq i \leq s/2\}.$$

In the next lemma, we show that every polytope symmetric about the origin contains a ball whose radius is determined by the facets of the polytope.

Lemma 2.2.17. *Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polyhedron symmetric about the origin,*

$$P = \{x \in \mathbb{R}^n | \langle x, h_i \rangle \leq 1 \textit{ and } \langle x, -h_i \rangle \leq 1 \textit{ for all } 1 \leq i \leq s/2\}.$$

Define $h := \min\{1/\|h_i\|_2 | 1 \leq i \leq s/2\}$. Then

$$\bar{B}_n^{(2)}(0, h) \subseteq P.$$

Proof. Every vector $x \in \bar{B}_n^{(2)}(0, h) = h \cdot \bar{B}_n^{(2)}(0, 1)$ is of the form

$$x = h \cdot x',$$

where $x' \in \mathbb{R}^n$ with $\|x'\|_2 \leq 1$. To show that x is contained in the polytope, we need to show that x satisfies all s constraints defining P.
For all $x \in \mathbb{R}^n$ with $1 \leq j \leq n$ we have

$$\langle h_j, x\rangle = h \cdot \langle h_j, x'\rangle \leq \frac{1}{\|h_j\|_2}\langle h_j, x'\rangle \leq \frac{1}{\|h_j\|_2} \cdot \|h_j\|_2 \cdot \|x'\|_2 \leq 1$$

using the Cauchy-Schwarz inequality. With the same argument, we see that $\langle -h_j, x\rangle \leq 1$. Altogether, this shows that $x \in P$. □

For the computation of the radius of a circumscribed Euclidean ball we need a result about the relation between the Euclidean length of a vector and its size and between the size of a matrix and its determinant.

Claim 2.2.18. *For $x \in \mathbb{Q}^n$ we have*

$$\|x\|_2 \leq \sqrt{n} \cdot \operatorname{size}(x).$$

For $D \in \mathbb{Q}^{n \times n}$ we have

$$|\det(D)| \leq n^{n/2} \operatorname{size}(D)^n.$$

Proof. The first statement follows directly from the definition of the Euclidean norm. For all $x \in \mathbb{R}^n$ we have

$$\|x\|_2^2 = \sum_{i=1}^n x_i^2 \leq \sum_{i=1}^n \operatorname{size}(x)^2 \leq n \cdot \operatorname{size}(x).$$

To prove the second statement we consider the columns $d_1, \dots, d_n$ of the matrix D. Using Hadamard's inequality, it follows directly from the first statement that

$$|\det(D)| \leq \prod_{i=1}^n \|d_i\|_2 \leq \prod_{i=1}^n \sqrt{n} \operatorname{size}(d_i) \leq \prod_{i=1}^n \operatorname{size}(D) = n^{n/2} \operatorname{size}(D)^n,$$

see Section A.0.1 in the Appendix. □

Using this result, we are able to compute the radius of a circumscribed Euclidean ball for every full-dimensional polytope. This radius is determined by the size of the polytope. First we compute the radius of a circumscribed ℓ_∞-ball for a given polytope.

Lemma 2.2.19. *Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope given by s integral inequalities $\langle a_i, x\rangle \leq \beta_i$, where $a_i \in \mathbb{Z}^n$, $\beta_i \in \mathbb{Z}$ for $1 \leq i \leq s$, i.e.,*

$$P = \{x \in \mathbb{R}^n | \langle a_i, x\rangle \leq \beta_i \text{ for } 1 \leq i \leq s\} = \left\{x \in \mathbb{R}^n | A^T x \leq \beta\right\},$$

where A is the matrix, which consists of the columns a_i. Then P is contained in an ℓ_∞-ball with radius $n^{n/2} r^n$, centered at the origin,

$$P \subseteq \bar{B}_n^{(\infty)}(0, n^{n/2} r^n),$$

where r is the representation size of the polytope.

Proof. Let $v \in P$ be an arbitrary vertex of the polytope. Then there exists a $n \times n$ submatrix C of A^T such that $C \cdot v = d$, where d is the column vector which consists of the corresponding coefficients of β. Using Cramer's Rule, the coefficients v_i of the vertex v are given by

$$v_i = \frac{\det(C_i)}{\det(C)},$$

where C_i is the matrix C, where the i-th column is replaced by d. Since A^T is a matrix with integer coefficients, we have $|\det(C)| \geq 1$ and we obtain for all coefficients of v the upper bound

$$|v_i| \leq |\det(C_i)| \leq n^{n/2}\,\text{size}(C)^n,$$

where the last inequality was shown in Claim 2.2.18. This proves that the statement of the lemma is correct. □

Corollary 2.2.20. *Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope given by m integral inequalities $\langle a_i, x\rangle \leq \beta_i$ where $a_i \in \mathbb{Z}^n$, $\beta_i \in \mathbb{Z}$ for $1 \leq i \leq m$, i.e.,*

$$P = \{x \in \mathbb{R}^n | \langle a_i, x\rangle \leq \beta_i \text{ for } 1 \leq i \leq m\} = \{x \in \mathbb{R}^n | A^T x \leq \beta\},$$

where A is the matrix which contains of the columns a_i. Then P is contained in an Euclidean ball with radius $n^{(n+1)/2} r^n$,

$$P \subseteq \bar{B}_n^{(2)}(0, R_{out}) \text{ with } R_{out} = n^{(n+1)/2} r^n$$

where r is an upper bound on the representation size of the polytope.

The proof of this statement follows directly from Lemma 2.2.19, using that it follows from Hölder's inequality that $\bar{B}_n^{(\infty)}(0,1) \subseteq \sqrt{n} \cdot \bar{B}_n^{(2)}(0,1)$.

Combining Corollary 2.2.20 and Lemma 7.1.4 we see that every full-dimensional polytope symmetric about the origin is a well-bounded convex body.

3. Lattices

In this chapter, we define several fundamental concepts and state important results from the geometry of numbers that will be used throughout this thesis. We focus on the interaction between lattices and convex sets and consider lattices from a purely mathematical point of view. For a more detailed introduction see [Cas71] or [MG02].

3.1. Fundamentals about lattices

A lattice L is a nonempty subset of $\mathbb{R}^n$ which is closed under addition and subtraction, i.e., if $v, w \in L$, then $v - w \in L$. Furthermore there exists an $\epsilon > 0$ such that the Euclidean ball $\bar{B}_n^{(2)}(0, \epsilon)$ does not contain a non-zero lattice vector.

Definition 3.1.1. *(Lattice)*
A lattice L is a discrete abelian subgroup of $\mathbb{R}^n$.

Each lattice has a basis, i.e., a sequence $b_1, \dots, b_m$ of m elements of L that generate the lattice as an abelian group. We denote this by $L = \mathcal{L}(B)$, where $B = [b_1, \dots, b_m] \in \mathbb{R}^{n \times m}$ is the matrix with the column vectors b_i. Then the lattice $\mathcal{L}(B)$ is the set of all linear integer combinations of the basis vectors, see Figure 3.1.

Definition 3.1.2. *(Lattice generated by a basis)*
Let $b_1, \dots, b_m \in \mathbb{R}^m$ be linearly independent (over $\mathbb{R}$). Set $B := [b_1, \dots, b_m] \in \mathbb{R}^{n \times m}$. The set

$$\mathcal{L}(B) := \left\{ \sum_{i=1}^{m} x_i b_i \middle| x_i \in \mathbb{Z} \text{ for } 1 \leq i \leq m \right\} = \{ Bx \mid x \in \mathbb{Z}^m \}$$

is called the lattice *generated by the basis vectors $b_1, \dots, b_m$.*

Definition 3.1.1 and Definition 3.1.2 are equivalent. That means, for each discrete (abelian) subgroup L of $\mathbb{R}^n$ there exists a basis $B \in \mathbb{R}^{n \times m}$ such that $L = \mathcal{L}(B)$ and each set $\mathcal{L}(B) \subseteq \mathbb{R}^n$ defined by m linearly independent vectors $B = [b_1, \dots, b_m]$ is a discrete abelian subgroup of $\mathbb{R}^n$, see for example [Bar02].

Obviously, there exist different bases that generate the same lattice. We call two matrices $B, B' \in \mathbb{R}^{n \times m}$ *equivalent* if they generate the same lattice, i.e., if $\mathcal{L}(B) = \mathcal{L}(B')$. Algebraically, two lattice bases are equivalent if and only if there exists a unimodular matrix $U \in \mathbb{Z}^{n \times n}$ such that $B' = B \cdot U$. A matrix $U \in \mathbb{Z}^{n \times n}$ is called *unimodular* if $|\det(U)| = 1$. The set of all unimodular matrices $U \in \mathbb{Z}^{n \times n}$ is called the special linear

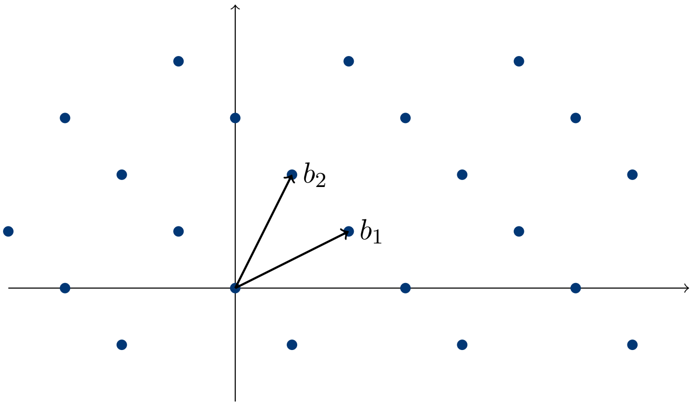

Figure 3.1.: **A lattice.** The lattice is generated by the vectors $b_1 = (1,2)^T \in \mathbb{R}^2$ and $b_2 = (2,1)^T \in \mathbb{R}^2$.

group over $\mathbb{Z}$ and is a multiplicative group.

A vector $v \in L$ is called *primitive* if $(1/k) \cdot v \notin L$ for all $k \in \mathbb{Z}\backslash\{0, \pm 1\}$. Every primitive lattice vector $v \in L$ can be extended to a basis of the lattice, i.e., there exists $b_2, \ldots, b_m \in L$ such that $\{v, b_2, \ldots, b_m\}$ is a basis of L. A proof of this result can be found for example in [Cas71].

If a lattice L is given by a basis $B \in \mathbb{R}^{n\times m}$, we call m the rank of L and n its dimension. If $m = n$, the lattice is full-dimensional. In this case, the vector space spanned by the basis, span(B), is the whole space $\mathbb{R}^n$. Obviously, the vector space spanned by a lattice is independent of the chosen basis. It is denoted by span(L). The dimension of span(L) corresponds to the rank of the lattice L.

Definition 3.1.3. *Let $L \subseteq \mathbb{R}^n$ be a lattice given by a basis $B \in \mathbb{R}^{n\times m}$. Then*

$$\operatorname{span}(L) := \left\{ \sum_{i=1}^{n} x_i b_i \mid x_i \in \mathbb{R} \text{ for } 1 \leq i \leq m \right\}$$

is the subspace of $\mathbb{R}^n$ which contains the lattice.

One can show that the subspace generated by a lattice $L \subseteq \mathbb{R}^n$ is the smallest subspace in $\mathbb{R}^n$ which contains the whole lattice.

For a set $B = [b_1, \ldots, b_m] \in \mathbb{R}^{n\times m}$ of linearly independent vectors we define the half open parallelepiped

$$\mathcal{P}(B) := \left\{ \sum_{j=1}^{n} \alpha_j b_j \mid 0 \leq \alpha_j < 1, j = 1, \ldots, n \right\}.$$

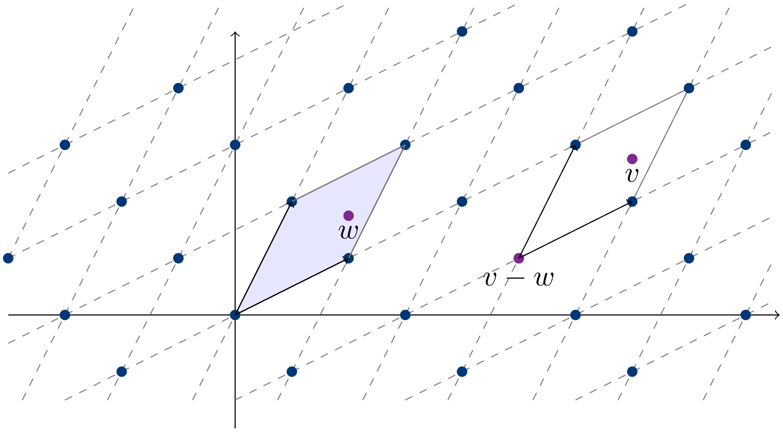

Figure 3.2.: **Fundamental parallelepiped.** The lattice is generated by the vectors $b_1 = (2,1) \in \mathbb{R}^2$ and $b_2 = (1,2) \in \mathbb{R}^2$. The corresponding fundamental parallelepiped covers the whole space $\mathbb{R}^2$. Thus for every vector $v \in \mathbb{R}^2$ there exists a unique vector $w \in \mathcal{P}(B)$ such that $v - w$ is a lattice vector.

If B is a basis of a lattice L, this parallelepiped is called *fundamental parallelepiped* or *fundamental region* of the lattice L with respect to the basis B. The fundamental parallelepiped can be used to define a disjoint covering of the whole subspace $\operatorname{span}(B)$,

$$\operatorname{span}(B) = \bigcup_{v \in \mathcal{L}(B)} (v + \mathcal{P}(B)),$$

as it is illustrated in Figure 3.2. If L is a full-dimensional lattice, we have $\operatorname{span}(B) = \mathbb{R}^n$ and the fundamental parallelepiped can be used to define a covering of the whole space.

Using a fundamental parallelepiped $\mathcal{P}(B)$ defined by some basis B of the lattice L, for every vector $v \in \operatorname{span}(B) = \operatorname{span}(L)$ we can define a unique representation $v = u + w$ with $u \in L$ and $w \in \mathcal{P}(B)$. We call two vectors v, w to be congruent modulo L, $v \equiv w \mod L$, if the difference vector $v - w$ is a lattice vector, i.e., if $v - w \in L$. By computing a vector w satisfying $w \equiv v \mod L$, we mean computing the unique $w \in \mathcal{P}(B)$ with $v - w \in L = \mathcal{L}(B)$.

Lemma 3.1.4. *Let $L \subseteq \mathbb{R}^n$ be a lattice given by a basis $B \in \mathbb{R}^{n \times m}$. For every vector $v \in \operatorname{span}(L)$ there exists a unique vector $w \in \mathcal{P}(B)$ such that $v - w \in L$.*

The proof of this result follows directly from the observation that for every vector $v \in \operatorname{span}(B) = \operatorname{span}(B)$ there exists a unique representation as a linear combination of the basis vectors, $v = \sum_{i=1}^{m} v_i b_i$ with $v_i \in \mathbb{R}$, $1 \leq i \leq m$. For all $1 \leq i \leq m$ the coefficients v_i can be uniquely represented as $v_i = \lfloor v_i \rfloor + (v_i - \lfloor v_i \rfloor)$, where $\lfloor v_i \rfloor \in \mathbb{Z}$ and $0 \leq v_i - \lfloor v_i \rfloor < 1$.

If we consider a lattice $L \subseteq \mathbb{R}^n$ of rank m together with m linearly independent lattice vectors, these lattice vectors do not necessarily form a basis of the lattice. A simple geometric criterion to decide whether a set of linearly independent lattice vectors $b_1', \ldots, b_m' \in L$ for a basis of the lattice is to consider the parallelepiped $\mathcal{P}(B')$ spanned by these vectors, where $B' := [b_1', \ldots, b_m']$. One can show that B' is a lattice basis of L if and only if the parallelepiped spanned by these vectors does not contain a non-zero lattice vector, i.e., if $\mathcal{P}(B') \cap L = \{0\}$.

For every set $B' = [b_1', \ldots, b_m']$ of m linearly independent lattice vectors of the lattice L the set $\mathcal{L}(B')$ is always a lattice and we have $\mathcal{L}(B') \subseteq L$. The lattice $\mathcal{L}(B')$ is called a *sublattice* of L. If $\mathcal{L}(B') \subsetneq L$ we say that $\mathcal{L}(B')$ is a proper sublattice of L.

Definition 3.1.5. *Let $L \subseteq \mathbb{R}^n$ be a lattice and $B' = [b_1', \ldots, b_m'] \in \mathbb{R}^{n \times m}$ with $b_i' \in L$ for all $1 \leq i \leq m$. For all $v \in L$ the set*

$$v + \mathcal{L}(B') := \{v + w | w \in \mathcal{L}(B')\}$$

is called a coset *of L modulo $\mathcal{L}(B')$. The set of all cosets is denoted by $L/\mathcal{L}(B')$.*

Determinant of a lattice

One fundamental constant of a lattice is its determinant.

Definition 3.1.6. *(Determinant of a lattice)*
Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m. The determinant of L, $\det(L)$, *is defined as the m-dimensional volume of the fundamental parallelepiped $\mathcal{P}(B)$ for some arbitrary basis $B \in \mathbb{R}^{n \times m}$ of L, i.e.,*

$$\det(L) := \sqrt{\det(B^T \cdot B)}.$$

The determinant of a lattice is a lattice invariant, i.e., it is independent of the basis defining the lattice. This follows directly from the observation that for two equivalent bases B, B' of a lattice L there exists a unimodular matrix U such that $B' = B \cdot U$ and we have $\det(B'^T B') = \det(B^T B)$.

If the lattice L is full-dimensional, we have $\det(L) = |\det(B)|$ and the determinant of the lattice is the n-dimensional volume of its fundamental parallelepiped. Geometrically, the inverse of the determinant of a lattice can be interpreted as the density of the lattice vectors.

Given a basis of the lattice the determinant can be computed in polynomial time using Gaussian elimination. Another way to compute the determinant of a lattice is to use the Gram-Schmidt orthogonalization of the basis.

Definition 3.1.7. *(Gram-Schmidt orthogonalization)*
Let $B = [b_1, \ldots, b_m] \in \mathbb{R}^{n \times m}$ be a set of linearly independent vectors. The Gram-Schmidt

orthogonalization $B^\dagger = [b_1^\dagger, \ldots, b_m^\dagger]$ *of* B *is defined by*

$$\begin{aligned} b_1^\dagger &:= b_1, \\ b_i^\dagger &:= b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^\dagger \textit{ for } 1 < i \leq n \textit{ and } \mu_{i,j} := \frac{\langle b_i, b_j^\dagger \rangle}{\langle b_j^\dagger, b_j^\dagger \rangle} \textit{ for } 1 \leq j < i \leq n. \end{aligned}$$

The Gram-Schmidt-orthogonalization depends on the order of the original basis vectors. The important properties of the Gram-Schmidt orthogonalization is that its vectors are pairwise orthogonal. Furthermore, they span the same vector space as the original vectors, that means we have $\operatorname{span}(b_1, \ldots, b_m) = \operatorname{span}(b_1^\dagger, \ldots, b_m^\dagger)$.

Given a basis $B \in \mathbb{R}^{n \times m}$ of a lattice L and the corresponding Gram-Schmidt orthogonalization $B^\dagger \in \mathbb{R}^{n \times m}$ we have $B = B^\dagger \cdot G$ where $G \in \mathbb{R}^{m \times m}$ is an upper triangular matrix whose elements on its diagonal are 1. Thus, we have

$$\begin{aligned} \det(B^T \cdot B) &= \det\left(G^T (B^\dagger)^T B^\dagger G\right) \\ &= \det(G^T) \cdot \det\left((B^\dagger)^T B^\dagger\right) \cdot \det(G) \\ &= \det\left((B^\dagger)^T B^\dagger\right) \end{aligned}$$

and it follows from Hadamard's inequality, that

$$\sqrt{\det(B^T B)} = \sqrt{\det\left((B^\dagger)^T B^\dagger\right)} = \prod_{i=1}^{m} \|b_i^\dagger\|_2,$$

see Section A.0.1 in the Appendix.

Lattices under orthogonal projection

Let $L \subseteq \mathbb{R}^n$ be a lattice given by a basis $[b_1, \ldots, b_m] \in \mathbb{R}^{m \times n}$. For simplicity, we assume that L is full-dimensional, i.e., $m = n$. Let L_k be the sublattice generated by the first k basis vectors, i.e.,

$$L_k := \mathcal{L}(b_1, \ldots, b_k)$$

for $1 \leq k \leq n$. For some fixed parameter k, $1 < k \leq n$, we define the mapping

$$\begin{aligned} \pi_k : \mathbb{R}^n &\longrightarrow \operatorname{span}(b_k^\dagger, \ldots, b_n^\dagger) \\ x &\mapsto \sum_{j=k}^{n} \frac{\langle x, b_j^\dagger \rangle}{\langle b_j^\dagger, b_j^\dagger \rangle} b_j^\dagger. \end{aligned}$$

Since $\mathbb{R}^n = \operatorname{span}(b_1^\dagger, \ldots, b_n^\dagger)$ and $\operatorname{span}(b_1, \ldots, b_{k-1}) = \operatorname{span}(b_1^\dagger, \ldots, b_{k-1}^\dagger)$, we have

$$\operatorname{span}(b_1, \ldots, b_{k-1})^\perp = \operatorname{span}(b_k^\dagger, \ldots, b_n^\dagger).$$

Thus, π_k is the orthogonal projection onto the orthogonal complement of $\mathrm{span}(L_{k-1})$, i. e., $\pi_k : \mathbb{R}^n \longrightarrow \mathrm{span}(L_{k-1})^\perp$.

We now consider the image of the lattice L under this orthogonal projection π_k and we define

$$L^{(n-k+1)} := \pi_k(L).$$

It is easy to see that $L^{(n-k+1)}$ is a lattice of rank $n-k+1$ and that a basis of this lattice is given by $[\pi_k(b_k), \ldots, \pi_k(b_n)]$. The lattice $L^{(n-k+1)}$ is often called the *projected lattice.* It is a classical technique to use this projected lattice to show statements by induction on the rank of the lattice.

Obviously, we have

$$\begin{aligned}\pi_k(b_j^\dagger) &= 0 \text{ for all } 1 \le j \le k-1 \text{ and}\\ \pi_k(b_j^\dagger) &= b_j^\dagger \text{ for all } k \le j \le n.\end{aligned}$$

Using this, we can show that the projection π_k decreases the length of a vector, that means for all $v \in \mathbb{R}^n$ we have $\|v\|_2 \ge \|\pi_k(v)\|_2$: Since the Gram-Schmidt orthogonalization $B^\dagger$ is a basis of $\mathbb{R}^n$, for every vector $v \in \mathbb{R}^n$ there exists a representation $v = \sum_{i=1}^n v_i b_i^\dagger$ with $v_i \in \mathbb{R}$ for all $1 \le i \le n$. Hence, the squared Euclidean length of v is at least

$$\begin{aligned}\|v\|_2^2 &= \sum_{i=1}^n v_i^2 \|b_i^\dagger\|_2^2 \ge \sum_{i=k}^n v_i^2 \|b_i^\dagger\|_2^2\\ &= \sum_{i=k}^n v_i^2 \|\pi_k(b_i^\dagger)\|_2^2 = \|\pi_k(v)\|_2^2.\end{aligned}$$

3.2. Minkowski's convex body theorem and successive minima

3.2.1. Minkowski's convex body theorem

Minkowski's convex body theorem provides a sufficient condition such that a convex body contains a lattice vector. It is based on a result of Blichfeldt which shows that every measurable set whose volume is larger than the determinant of the lattice contains a non-zero lattice vector. The proof of Blichfeldt's theorem uses a generalization of the pigeonhole principle.

Theorem 3.2.1. *(Blichfeldt's Theorem)*
Let $L \subseteq \mathbb{R}^n$ be a full-dimensional lattice and $\mathcal{S} \subseteq \mathbb{R}^n$ be a measurable set. If $\mathrm{vol}_n(\mathcal{S}) >$ $\det(L)$*, there exists two vectors $z_1, z_2 \in \mathcal{S}$, $z_1 \neq z_2$, such that $z_1 - z_2 \in L$.*

Proof. Let $B \in \mathbb{R}^{n \times n}$ be a basis of the lattice L. Then the set $L + \mathcal{P}(B)$ is a partition of the space $\mathbb{R}^n$, that means $L + \mathcal{P}(B) = \mathbb{R}^n$ and for all $u, v \in L$, $u \neq v$, we have $u + \mathcal{P}(B) \cap v + \mathcal{P}(B) = \emptyset$.
For each lattice vector $v \in L$ we consider the intersection of the set $\mathcal{S}$ with the corresponding translation of the fundamental parallelepiped,

$$\mathcal{S}_v := \mathcal{S} \cap (v + \mathcal{P}(B)).$$

Since B is a basis of $\mathbb{R}^n$ and the lattice is full-dimensional, the sets $\mathcal{S}_v$, $v \in L$, are a partition of the set $\mathcal{S}$ and we have

$$\operatorname{vol}_n(\mathcal{S}) = \sum_{v \in L} \operatorname{vol}_n(\mathcal{S}_v).$$

Now we consider the translations of the sets $\mathcal{S}_v$ in the fundamental parallelepiped. For $v \in L$ we define

$$\mathcal{S}'_v := \mathcal{S}_v - v.$$

Then we have for all $v \in L$ that $\mathcal{S}'_v = (\mathcal{S} - v) \cap \mathcal{P}(B)$ and $\operatorname{vol}_n(\mathcal{S}'_v) = \operatorname{vol}_n(\mathcal{S}_v)$. It follows that

$$\sum_{v \in L} \operatorname{vol}_n(\mathcal{S}'_v) = \sum_{v \in L} \operatorname{vol}_n(\mathcal{S}_v) = \operatorname{vol}_n(S) > \det(L) = \operatorname{vol}_n(\mathcal{P}(B)).$$

Since for all $v \in L$ we have $\mathcal{S}'_v \subseteq \mathcal{P}(B)$, this shows that there exists vectors $v, w \in L$ such that $\mathcal{S}'_v \cap \mathcal{S}'_w \neq \emptyset$. Let $z \in \mathcal{S}'_v \cap \mathcal{S}'_w$. Since $z \in \mathcal{S}'_v$ there exists $z_1 \in \mathcal{S}_v \subseteq \mathcal{S}$ such that $z = z_1 - v$. Since $z \in \mathcal{S}'_w$ there exists $z_2 \in \mathcal{S}_w \subseteq \mathcal{S}$ such that $z = z_2 - w$. Obviously, we have $v \neq w$ and $z_1 \neq z_2$. From this it follows

$$z_1 - z_2 = z + v - (z + w) = v - w \in L \backslash \{0\}.$$

□

As a consequence, we obtain

Theorem 3.2.2. *(Minkowski's convex body theorem)*
Let $L \subseteq \mathbb{R}^n$ be a full-dimensional lattice and $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set, which is symmetric about the origin. If $\operatorname{vol}_n(\mathcal{C}) > 2^n \det(L)$, the set $\mathcal{C}$ contains a non-zero lattice vectors $v \in \mathcal{C} \cap L \backslash \{0\}$.

Proof. We consider the set $\mathcal{C}' := (1/2) \cdot \mathcal{C}$. The volume of $\mathcal{C}'$ satisfies

$$\operatorname{vol}_n(\mathcal{C}') = 2^{-n} \operatorname{vol}_n(\mathcal{C}) > \det(L),$$

see Lemma 2.1.3 in Chapter 2. According to Blichfeldts' Theorem, Theorem 3.2.1, there exists two vectors $z_1, z_2 \in \mathcal{C}'$ such that $z_1 - z_2 \in L \backslash \{0\}$.
Furthermore, we have $2z_1, 2z_2 \in \mathcal{C}$ and due to the symmetry of $\mathcal{C}$ it follows that $-2z_2 \in \mathcal{C}$. Since $\mathcal{C}$ is convex, we have

$$\frac{1}{2}(2z_1 + (-2z_2)) = z_1 - z_2 \in \mathcal{C}.$$

This shows that $\mathcal{C}$ contains the non-zero lattice vector $z_1 - z_2$. □

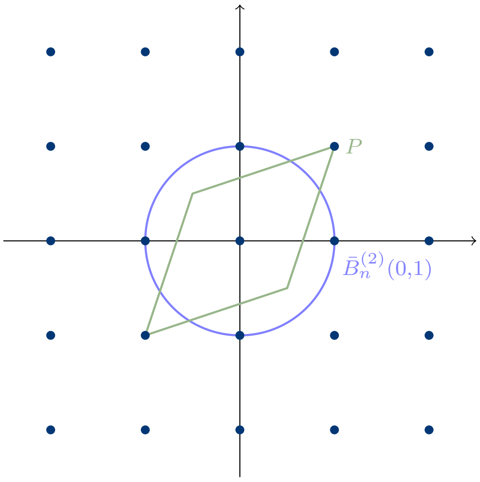

Figure 3.3.: **The minimum distance of a lattice in different norms.** We consider the integer lattice $\mathbb{Z}^2$. With respect to the Euclidean norm, the unit vectors $\pm e_1$ and $\pm e_2$ are the shortest non-zero lattice vectors in $\mathbb{Z}^2$. If we consider the norm defined by the polytope P symmetric about the origin, we see that the vectors $(1,1) \in \mathbb{Z}^2$ and $(-1,-1) \in \mathbb{Z}^2$ are the shortest non-zero lattice vectors.

3.2.2. Successive minima

A fundamental parameter of a lattice is its minimal distance which is defined as the minimal distance between two different lattice vectors

$$\min\{\|x-y\| | x, y \in L, x \neq y\}.$$

This distance can be defined for any norm $\|\cdot\|$ on $\mathbb{R}^n$. Obviously, the minimal distance between two different lattice vectors is the same as the length of a shortest non-zero lattice vector: Since a lattice is a subgroup of $\mathbb{R}^n$, it is closed under addition and subtraction. Thus, the difference vector $x-y$ of two distinct lattice vectors $x, y \in L$ is guaranteed to be a non-zero lattice vector. We define

$$\lambda_1^{(\|\cdot\|)}(L) := \min\{\|x-y\| | x, y \in L, x \neq y\}.$$

and call it the *first successive minimum of the lattice L with respect to the norm $\|\cdot\|$*. The first successive minimum of a lattice is the length of the shortest non-zero vector in the lattice. Obviously, the minimum distance of a lattice and the vector achieving it depends on the corresponding norm as it is illustrated in Figure 3.3.

The number of shortest non-zero lattice vectors with respect to the Euclidean norm is called the *kissing number* of the lattice. One can show that this number is at most single

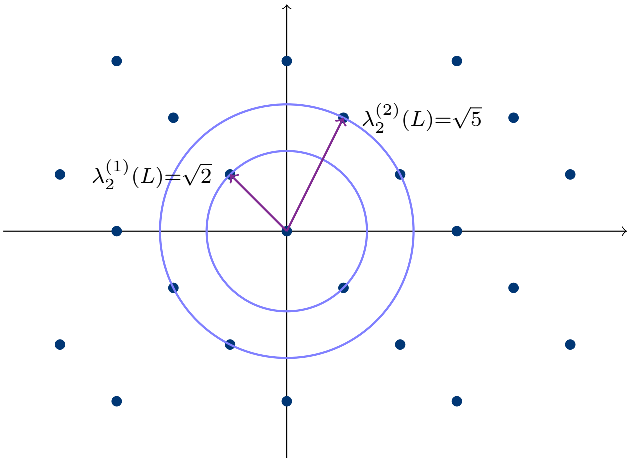
$\lambda_2^{(2)}(L){=}\sqrt{5}$
$\lambda_2^{(1)}(L){=}\sqrt{2}$

Figure 3.4.: **The successive minima of a lattice.** The lattice is generated by the vectors $(1,2)^T \in \mathbb{R}^2$ and $(2,1)^T \in \mathbb{R}^2$. The minimum distance of this lattice with respect to the Euclidean norm of this lattice is $\sqrt{2}$ and the length of the second successive minimum is $\sqrt{5}$.

exponential in the dimension, see [CS93].

An equivalent way to define the first successive minimum $\lambda_1^{(\|\cdot\|)}(L)$ is the following: $\lambda_1^{(\|\cdot\|)}(L)$ is the radius of the smallest ball centered in the origin containing one linearly independent lattice vector.
This definition can be generalized easily to define the i-th successive minimum of a lattice with $1 \leq i \leq n$ as the smallest real number ρ such that L contains i linearly independent vectors of length at most ρ, see Figure 3.4.

Definition 3.2.3. *(Successive minima)*
Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m and $\|\cdot\|$ be a norm on $\mathbb{R}^n$. The i-th successive minimum $\lambda_i^{(\|\cdot\|)}(L)$ *is defined as*

$$\lambda_i^{(\|\cdot\|)}(L) := \inf\left\{\rho \;\middle|\; \dim\left(\operatorname{span}(L) \cap \bar{B}_n^{(\|\cdot\|)}(0,\rho)\right) \geq i\right\}.$$

Since every lattice L is discrete, there exists a constant $\epsilon > 0$ such that the minimal distance of this lattice is at least ϵ. From this it follows that for every $\rho > 0$, the ball $\bar{B}_n^{(\|\cdot\|)}(0,\rho)$ around the origin with radius ρ contains only finitely many lattice vectors. Using this, one can show that for every lattice L of rank m, there exist linearly independent lattice vectors whose lengths are exactly the successive minima, i.e., there exists $v_1, \ldots, v_m \in L$ linearly independent with $\|v_i\| = \lambda_i^{(\|\cdot\|)}(L)$, see [Cas71]. It is not guaranteed that these vectors are a basis of the lattice.

Using Minkowski's convex body theorem, it can be guaranteed that every lattice contains a non-zero lattice vector whose length with respect to the ℓ_∞-norm is at most $\det(L)^{1/m}$, where m is the rank of the lattice.

Theorem 3.2.4. *Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m. Then*

$$\lambda_1^{(\infty)}(L) \leq \det(L)^{1/m}.$$

Proof. Obviously, the intersection of the open ℓ_∞-ball with radius $\lambda_1^{(\infty)}(L)$ with $\operatorname{span}(L)$ contains exactly one lattice vector,

$$\left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right) \cap L = \{0\}.$$

Let $B \in \mathbb{R}^{n\times m}$ be a basis of the lattice L. Then there exists a rotation given by an orthogonal matrix $O \in \mathbb{R}^{n\times n}$ such that

$$\operatorname{span}(O \cdot B) = \operatorname{span}(e_1, \dots, e_m) = \mathbb{R}^n \cap \bigcap_{i=m+1}^{n} H_{0,e_i}.$$

Obviously, the convex set $O \cdot \left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right)$ contains exactly one lattice vector from the lattice $\mathcal{L}(O \cdot B)$,

$$O \cdot \left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right) \cap \mathcal{L}(O \cdot B) = \{0\}.$$

Since $\mathcal{L}(O \cdot B) \subseteq \mathbb{R}^n \cap \bigcap_{i=m+1}^{n} H_{0,e_i}$ and $O \cdot \left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right) \subseteq \mathbb{R}^n \cap \bigcap_{i=m+1}^{n} H_{0,e_i}$ we can apply Minkowski's convex body theorem, Theorem 3.2.2, and obtain that

$$\operatorname{vol}_m\left(O \cdot \left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right)\right) \leq 2^m \det(\mathcal{L}(OB)). \tag{3.1}$$

Since O is an orthogonal matrix, we have $\det(\mathcal{L}(OB)) = \det(\mathcal{L}(B)) = \det(L)$ and

$$\begin{aligned}\operatorname{vol}_m\left(O \cdot \left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right)\right) &= \operatorname{vol}_m\left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right) \\ &= \lambda_1^{(\infty)}(L)^m \cdot \operatorname{vol}_m\left(B_n^{(\infty)}(0, 1) \cap \operatorname{span}(L)\right).\end{aligned}$$

One can show that the m-dimensional volume of the intersection of the n-dimensional ℓ_∞-unit ball with an m-dimensional subspace has volume at least 2^m, that means for every m-dimensional subspace $H \subseteq \mathbb{R}^n$ we have

$$\operatorname{vol}_m(B_n^{(\infty)}(0, 1) \cap H) \geq 2^m.$$

This result is shown by Vaaler, see [Vaa79]. Thus, we obtain that

$$\operatorname{vol}_m\left(O \cdot \left(B_n^{(\infty)}(0, \lambda_1^{(\infty)}(L)) \cap \operatorname{span}(L)\right)\right) \geq \left(2\lambda_1^{(\infty)}(L)\right)^m.$$

Combining this with (3.1) we obtain

$$\lambda_1^{(\infty)}(L) \leq \det(L)^{1/m}.$$

□

Unfortunately, the proof of this result is not constructive. It is easy to see that this bound is tight if we consider the integer lattice which satisfies $\lambda_1^{(\infty)}(\mathbb{Z}^n) = 1 = \det(\mathbb{Z}^n)^{1/n}$. Using Hölder's inequality, we obtain corresponding results for arbitrary ℓ_p-norms. The result for the Euclidean norm is also known as Minkowski's first theorem.

Corollary 3.2.5. *(Minkowski's first theorem)*
Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m. Then

$$\lambda_1^{(2)}(L) \leq \sqrt{n} \cdot \det(L)^{1/m}.$$

It can be shown that this result is asymptotically tight, i.e., there exist lattices $L \subseteq \mathbb{R}^n$ such that $\lambda_1^{(2)}(L) > c \cdot \sqrt{n} \det(L)^{1/n}$ for some fixed constant c. Minkowski also proved a stronger result which gives an upper bound on the product of all successive minima of a lattice. For a proof of the following theorem see [Mar03].

Theorem 3.2.6. *(Minkowski's second theorem)*
Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m. Then

$$\left(\prod_{i=1}^{m} \lambda_i^{(2)}(L) \right)^{1/m} \leq \sqrt{n} \det(L)^{1/m}.$$

3.2.3. Packing radius and covering radius

There exists a further geometric interpretation of the minimum distance of a lattice: We consider balls around each lattice vector. Then $\lambda_1^{(\|\cdot\|)}(L)/2$ is the largest number α such that the open balls $B_n^{(\|\cdot\|)}(v, \alpha)$ with $v \in L$ do not intersect. We have

$$B_n^{(\|\cdot\|)}\left(v, \frac{\lambda_1^{(\|\cdot\|)}(L)}{2}\right) \bigcap B_n^{(\|\cdot\|)}\left(w, \frac{\lambda_1^{(\|\cdot\|)}(L)}{2}\right) = \emptyset \text{ for all } v, w \in L,\ v \neq w$$

and for all $\alpha > \lambda_1^{(\|\cdot\|)}(L)/2$ there exist $v, w \in L$, $v \neq w$, such that

$$B_n^{(\|\cdot\|)}(v, \alpha) \cap B_n^{(\|\cdot\|)}(w, \alpha) \neq \emptyset.$$

The value $\lambda_1^{(\|\cdot\|)}(L)/2$ is called the *packing radius* of the lattice.
Compared with this, the *covering radius* of a lattice is the smallest radius such that the closed balls centered at all lattice vectors cover the whole space. For an illustration see Figure 3.5.

Definition 3.2.7. *(Covering radius)*
Let $L \subseteq \mathbb{R}^n$ be a lattice. The covering radius *of L with respect to the norm $\|\cdot\|$ is the smallest radius ρ such that the balls $\bar{B}_n^{(\|\cdot\|)}(v, \rho)$, $v \in L$, cover the whole space* span(L)*, i.e.,*

$$\operatorname{span}(L) = \bigcup_{v \in L} \bar{B}_n^{(\|\cdot\|)}(v, \rho).$$

The covering radius is denoted by $\mu^{(\|\cdot\|)}(L)$.

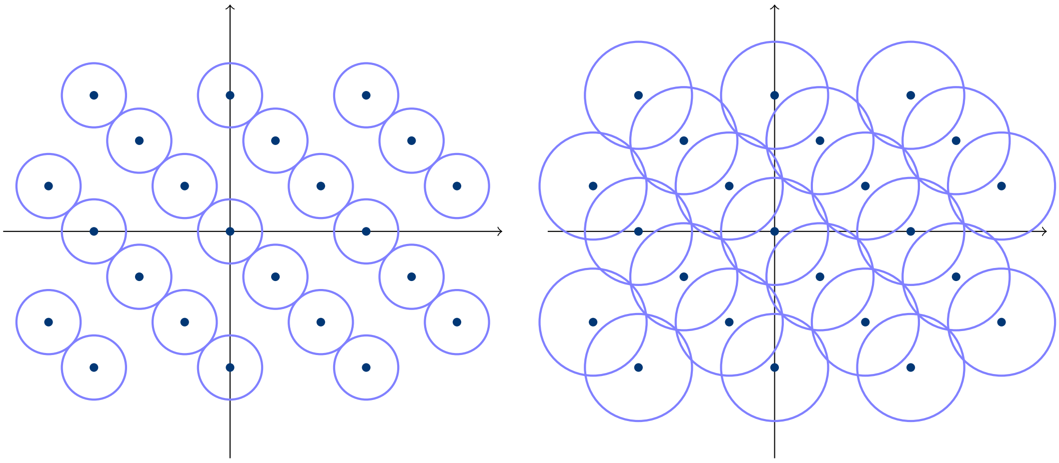

Figure 3.5.: **The packing radius and the covering radius of a lattice.** We consider the lattice L generated by $b_1 = (2,1)^T \in \mathbb{R}^2$ and $b_2 = (1,2)^T \in \mathbb{R}^2$. The left side shows a packing of the lattice L with Euclidean balls with radius $\lambda_1^{(2)}(L)/2$, whereas the right side shows a covering of the lattice with Euclidean balls with radius $\mu^{(2)}(L)$.

3.3. The dual lattice and transference bounds

Analogously as the dual space of a vector space, we can consider the dual of a lattice $L \subseteq \mathbb{R}^n$. The dual of a lattice L is the set

$$L^* = \{f : L \to \mathbb{Z} | f \text{ linear}\} \tag{3.2}$$

of all functions $f : L \to \mathbb{Z}$ satisfying $f(\alpha \cdot v + \beta \cdot w) = \alpha \cdot f(v) + \beta \cdot f(w)$ for all $\alpha, \beta \in \mathbb{R}$, $v, w \in L$.

One can show that L and L^* are isomorphic by defining an isomorphism from the group L to the group L^*. Given a basis $B \in \mathbb{R}^{n \times m}$, $B = [b_1, \ldots, b_m]$ of the lattice L, every lattice vector $v \in L$ can be represented as an integer linear combination of the basis vectors, $v = \sum_{i=1}^m v_i b_i$ with $v_i \in \mathbb{Z}$ for all $1 \le i \le m$. Thus, the functions $\beta_i : L \to \mathbb{Z}$, $\sum_{j=1}^m v_j b_j \mapsto v_j$ are well-defined linear functions. Now, it is easy to see that the mapping $L \to L^*$, $\sum_{i=1}^m v_i b_i \to \sum_{i=1}^m v_i \cdot \beta_i$ is an isomorphism.

The definition of a dual lattice as it is given in (3.2) is not very useful in practice. It is more common to represent the elements of the dual lattice by vectors. Every vector $v \in \operatorname{span}(L)$ can be interpreted as the linear map

$$f_v : L \to \mathbb{R}, \ y \mapsto \langle v, y \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean scalar product. However, f_v does not need to be an element in L^* since $\langle v, y\rangle$ is not mandatory an element in $\mathbb{Z}$. Moreover, we have

$$f_v \in L^* \text{ if and only if } \langle v, y\rangle \in \mathbb{Z} \text{ for all } y \in L.$$

This leads to the following equivalent definition of a dual lattice.

Definition 3.3.1. *(Dual lattice)*
Let $L \subseteq \mathbb{R}^n$ be a lattice. The dual lattice L^* of the lattice L is defined as the set

$$L^* = \{x \in \operatorname{span}(L) | \langle x, v\rangle \in \mathbb{Z} \textit{ for all } v \in L\}.$$

Thus the dual lattice L^* is the set of all vectors in the vector space spanned by the lattice L, whose scalar product with every lattice vector is an integer.

Obviously, if $B = [b_1, \ldots, b_m]$ is a basis of the lattice L, we observe that for a vector $y \in \operatorname{span}(L)$ we have that $\langle v, y\rangle \in \mathbb{Z}$ for all $v \in L$ if and only if $\langle v, b_i\rangle \in \mathbb{Z}$ for all $1 \leq i \leq n$. Hence, $L^* = \{y \in \operatorname{span}(L) | \langle y, b_i\rangle \in \mathbb{Z} \text{ for all } 1 \leq i \leq n\}$.

The integer lattice $\mathbb{Z}^n$ is self dual, i.e., $(\mathbb{Z}^n)^* = \mathbb{Z}^n$ since for all $v, w \in \mathbb{Z}^n$ we have $\langle v, w\rangle \in \mathbb{Z}$.

Before we state some important properties of the dual lattice, we give a geometric interpretation of it.

3.3.1. Geometric representation of the dual lattice

Let $L = \mathcal{L}(B) \subseteq \mathbb{R}^n$ be a lattice of rank m given by a basis $B = [b_1, \ldots, b_m]$. As we have seen, the corresponding dual lattice consists of all vectors in span(B) whose scalar product with all basis vectors is an integer.

We start with the first basis vector b_1. The set of all vectors in span(B) whose scalar product with b_1 is zero is the hyperplane

$$H_{0,b_1} = \{y \in \operatorname{span}(B) | \langle y, b_1\rangle = 0\}$$

orthogonal to b_1.
Let $v_1 \in \operatorname{span}(B)$ be a vector whose scalar product with b_1 is exactly 1, for example $v_1 = b_1/\|b_1\|_2^2$. The translations of the hyperplane H_{0,b_1} in direction of this vector are the affine hyperplanes $r \cdot v_1 + H_{0,b_1}$ with $r \in \mathbb{R}$. For a fixed number r the affine hyperplane $r \cdot v_1 + H_{0,b_1}$ consists of all vectors in span(B) whose scalar product with b_1 is exactly r. Hence, all vectors in span(B) which have an integer scalar product with b_1 are contained in an affine hyperplane of the form

$$\begin{aligned} H_{k,b_1} &= \{y \in \operatorname{span}(L) | \langle b_1, y\rangle = k\} \\ &= \{k \cdot v_1 + y | y \in \operatorname{span}(B) \text{ satisfying } \langle y, b_1\rangle = 0\} \\ &= k \cdot v_1 + H_{0,b_1}, \end{aligned}$$

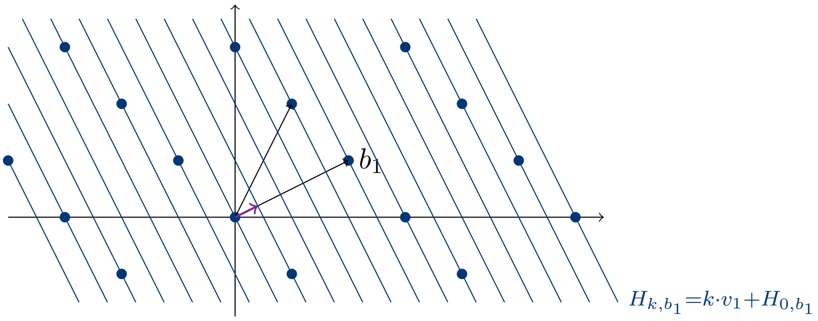

Figure 3.6.: **Construction of the dual lattice (I).** The lattice is generated by the vectors $b_1 = (2,1)^T \in \mathbb{R}^2$ and $b_2 = (1,2)^T \in \mathbb{R}^2$. The vector $v_1 = (1/5) \cdot (1,1)^T \in \mathbb{R}^2$ satisfies $\langle v_1, b_1 \rangle = 1$. We observe that there exist affine hyperplanes which do not contain lattice vectors.

where $k \in \mathbb{Z}$. That means we have

$$\{y \in \operatorname{span}(L) | \langle y, b_1 \rangle = k\} = \bigcap_{k \in \mathbb{Z}} H_{k,b_1}.$$

The Euclidean distance between the distinct affine hyperplanes is

$$\|v_1\|_2 = \frac{\|b_1\|_2}{\|b_1\|_2^2} = \frac{1}{\|b_1\|_2}. \tag{3.3}$$

That means the longer the Euclidean length of the vector b_1 is, the shorter the distance between the distinct affine hyperplanes. The whole situation is illustrated in Figure 3.6.

We can proceed in the same way for the other basis vectors. Then the dual lattice is the set of all intersections of the different affine hyperplanes as it is illustrated in Figure 3.7.

3.3.2. Properties of the dual lattice

Definition 3.3.2. *Let $B \in \mathbb{R}^{n \times m}$ be a lattice basis of the lattice L. Then, the corresponding dual basis $D \in \mathbb{R}^{n \times m}$ is defined by*

- $\operatorname{span}(D) = \operatorname{span}(B)$ *and*
- $B^T \cdot D = I_n$.

As the solution of a linear equation system the matrix D is uniquely determined, i.e. for every lattice basis B there exists a uniquely determined dual basis. If the lattice $L = \mathcal{L}(B)$ is full-dimensional, then $D = (B^T)^{-1}$. Otherwise, we have $D = B(B^T \cdot B)^{-1}$.

Furthermore, we can show that $L^* = \mathcal{L}(D)$ is a lattice and that $(L^*)^* = L$.

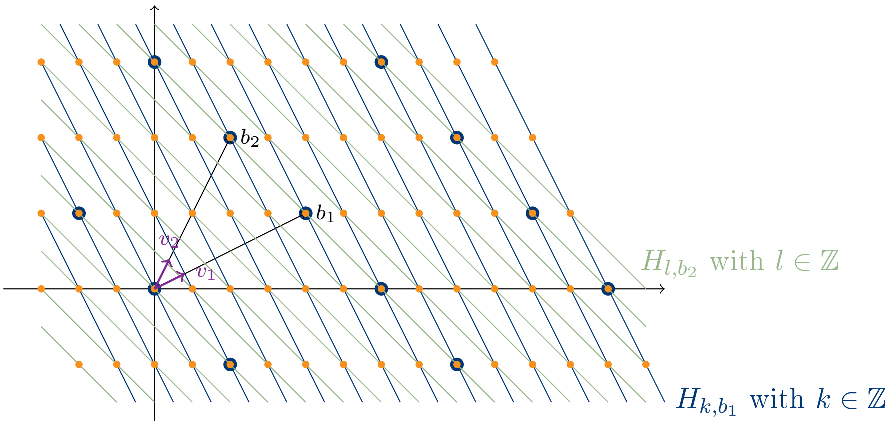
b_2
b_1
v_2
v_1
H_{l,b_2} with $l \in \mathbb{Z}$
H_{k,b_1} with $k \in \mathbb{Z}$

Figure 3.7.: **Construction of a dual lattice (II).** The lattice is generated by the vectors $b_1 = (2,1)^T \in \mathbb{R}^2$ and $b_2 = (1,2)^T \in \mathbb{R}^2$. The dual lattice is the set of all intersection points of the affine hyperplanes H_{k,b_1} and H_{l,b_2} with $k, l \in \mathbb{Z}$.

The determinant of the dual lattice is the inverse of the determinant of the original lattice. This follows directly from the linearity of the determinant.

Lemma 3.3.3. *Let $L \subseteq \mathbb{R}^n$ be a lattice. Then,*

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

For a vector $v \in \mathbb{R}^n \backslash \{0\}$ the hyperplane $H_{0,v} = \{x \in \mathbb{R}^n | \langle x, v\rangle = 0\}$ is a subspace of the vector space $\mathbb{R}^n$ of dimension $n-1$ which consists of all vectors orthogonal to v. For lattices, we can show a similar result using the primitive vectors in the dual lattice. We can show that for every primitive vector $v \in L^*$ the intersection $L \cap H_{0,v}$ is a sublattice of L of rank $n-1$.

Lemma 3.3.4. *Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m and $v \in L^*$ be a primitive vector. Then $L \cap H_{0,v}$ is a sublattice of L of rank $m-1$.*

Proof. Since $v \in L^*$ is a primitive vector, there exists a lattice basis of L^* which contains v, $V = [v, v_2, \ldots, v_m] \subseteq L^*$. Let $B = [b_1, \ldots, b_m]$ be the corresponding dual basis. That means we have

$$\langle v, b_i\rangle = 0 \text{ for all } 2 \leq i \leq n.$$

This shows that the sublattice $\mathcal{L}(b_2, \ldots, b_m)$ of rank $m-1$ is contained in $L \cap H_{0,v}$. □

This result shows that a primitive vector $v \in L^*$ can be used to represent a lattice as the union of sets

$$L = \bigcup_{k \in \mathbb{Z}} L_{k,v} \text{ with } L_{k,v} := \{x \in L | \langle x, v\rangle = k\},$$

where the sets $L_{k,v}$ are translations of the sublattice $L_{0,v} = L \cap H_{0,v}$, i.e.,

$$L_{k,v} = L_{0,v} + k\frac{1}{\langle v, v\rangle}v.$$

3.3.3. Transference bounds

The relevance of the dual lattice is that some important informations about a given lattice L can be extracted from the properties of the corresponding dual lattice L^*. These relations are described in so-called transference bounds. For example the following relation is a direct applications of Minkowski's first theorem.

Lemma 3.3.5. *Let $L \subseteq \mathbb{R}^n$ be a lattice and $L^* \subseteq \mathbb{R}^n$ be the corresponding dual lattice. Then,*

$$\lambda_1^{(2)}(L) \cdot \lambda_1^{(2)}(L^*) \leq n.$$

Proof. Using Minkowski's first theorem, Corollary 3.2.5, we obtain that

$$\lambda_1^{(2)}(L) \leq \sqrt{n}\det(L)^{1/n} \text{ and}$$
$$\lambda_1^{(2)}(L^*) \leq \sqrt{n}\det(L^*)^{1/n}.$$

Since $\det(L) = 1/\det(L^*)$, the statement follows. □

The following theorem is a remarkable result relating the covering radius of a lattice and the minimum distance of the corresponding dual lattice.

Theorem 3.3.6. *(Banszczyk, [Ban93])*
Let $L \subseteq \mathbb{R}^n$ be a lattice and $L^* \subseteq \mathbb{R}^n$ be the corresponding dual lattice. Then we have

$$1 \leq \mu^{(2)}(L) \cdot \lambda_1^{(2)}(L^*) \leq \frac{n}{2}.$$

This result was proven by Banaszczyk in 1993 using techniques from harmonic analysis. It can be generalized to non-Euclidean norms using the so-called dual norm, see [BLPS99].

To illustrate the geometric interpretation of this transference bound of Banaszczyk we suppose that we are given a lattice $L \subseteq \mathbb{R}^n$ where the covering radius of the lattice is greater than ρ, $\mu^{(2)}(L) > \rho$. Now it follows from the transference bound stated in Theorem 3.3.6 that the minimum distance of the dual lattice is upper bounded by

$$\lambda_1^{(2)}(L^*) \leq \frac{\mu^{(2)}(L)}{\rho} \cdot \lambda_1^{(2)}(L^*) < \frac{n}{2\rho}.$$

Thus, if $v \in L^*$ is a shortest non-zero lattice vector in L^* and we consider the representation of L as $L = \bigcup_{k\in\mathbb{Z}} L_{k,v}$, then each translation $L_{k,v}$ is contained in the affine hyperplane $H_{k,v}$ and the Euclidean distance between the affine hyperplanes $H_{k,v}$, $k \in \mathbb{Z}$, is at least $2\rho/n$, see Equation (3.3). In other words, the translations $L_{k,v}$ are well-separated.

For a more detailed introduction into the algorithmic use of the dual lattice see [Vaz01].

4. Lattices: A complexity theoretic perspective

In this chapter we consider lattices from the complexity theoretical point of view. We give a formal definition of the four classical lattice problems from the geometry of numbers, the shortest vector problem (SVP), the successive minima problem (SMP), the shortest independent vectors problem (SIVP), and the closest vector problem (CVP). We state the most important results concerning their complexity and we present the main algorithms that solve these problems.

After these general considerations, we take a closer look on the lattice problems. First of all, we observe why it is (mostly) difficult to adapt an algorithm that solves a lattice problem with respect to the Euclidean norm to an algorithm that solves the corresponding lattice problem with respect to an arbitrary norm. Furthermore, we consider the number of possible solutions of the four lattice problems and we will see why the solution of SVP is comparatively uncomplicated compared with the solution of SMP, SIVP, and CVP.

Then we focus on the relation between SVP, SMP, SIVP, and CVP to develop approaches for a unified algorithmic treatment of these problems. Based on these results we will present in Chapter 5 and Chapter 6 algorithms for all four lattice problems for arbitrary norms, in particular for ℓ_p-norms with $1 \leq p \leq \infty$.

4.1. The lattice problems SVP, SMP, SIVP, and CVP

In the following we consider some arbitrary norm $\|\cdot\|$ on $\mathbb{R}^n$. We start with a formal definition of the shortest vector problem which is associated to the minimum distance of a lattice.

Definition 4.1.1. *(Shortest Vector Problem (SVP))*
Given a lattice $L \subseteq \mathbb{R}^n$, find a non-zero lattice vector $v \in L\backslash\{0\}$ such that

$$||v|| = \lambda_1^{(\|\cdot\|)}(L),$$

i.e., $\|v\| \leq \|w\|$ for any other $w \in L\backslash\{0\}$.

This variant of the shortest vector problem is also denoted as the search version of the shortest vector problem since the goal is really to find a shortest non-zero lattice vector. There are two other variants of the shortest vector problem, the optimization variant and the decisional variant. In the *optimization shortest vector problem* we are given a lattice and the goal is to determine the minimum distance of the lattice with respect to the corresponding norm. In the *decisional shortest vector problem* we are given a lattice and an additional parameter $\alpha > 0$. Here, the goal is to decide whether the minimum distance of the lattice is at most α. Kannan showed that all three versions of the shortest vector problem are polynomial time equivalent, see [Kan87b].

Often we are not able to solve the shortest vector problem exactly. Thus we consider an approximated version of the shortest vector problem and look for approximation algorithms for SVP.

Definition 4.1.2. *(γ-Approximate Shortest Vector Problem (SVP$_\gamma$))*
Given a lattice $L \subseteq \mathbb{R}^n$, find a vector $v \in L\backslash\{0\}$ such that $\|v\| \leq \gamma \cdot \lambda_1^{(\|\cdot\|)}(L)$.

The parameter $\gamma \geq 1$ is some arbitrary approximation factor. The approximation factor can be a constant or a function of any parameter associated to the lattice. Often the parameter γ depends on the dimension of the lattice. For the other variants of SVP approximate versions can be defined analogously.

As the minimum distance of a lattice can be generalized to the successive minima of a lattice, we can generalize the problem to compute a shortest non-zero lattice vector to the problem to compute n linearly independent lattice vectors with minimal length.

Definition 4.1.3. *(Successive Minima Problem (SMP))*
Given a lattice $L \subseteq \mathbb{R}^n$ of rank m, find m linearly independent vectors $v_1, \ldots, v_m \in L$ such that

$$||v_i|| = \lambda_i^{(\|\cdot\|)}(L)$$

for all $i = 1, \ldots, m$.

As we have already seen in Chapter 3, every lattice contains linearly independent vector achieving the successive minima. In many situations it is not important to compute m linearly independent lattice vectors where each vector is as short as possible but to compute m linearly independent lattice vectors where all vectors are not too long. The task to compute such vectors is called the shortest independent vectors problem.

Definition 4.1.4. *(Shortest Independent Vectors Problem (SIVP))*
Given a lattice $L \subseteq \mathbb{R}^n$ of rank m, find m linearly independent vectors $v_1, \ldots, v_m \in L$ such that

$$||v_i|| \leq \lambda_m^{(\|\cdot\|)}(L)$$

for all $i = 1, \ldots, m$.

Analogously as in the case of the shortest vector problem, we can define approximate versions of SMP and SIVP.

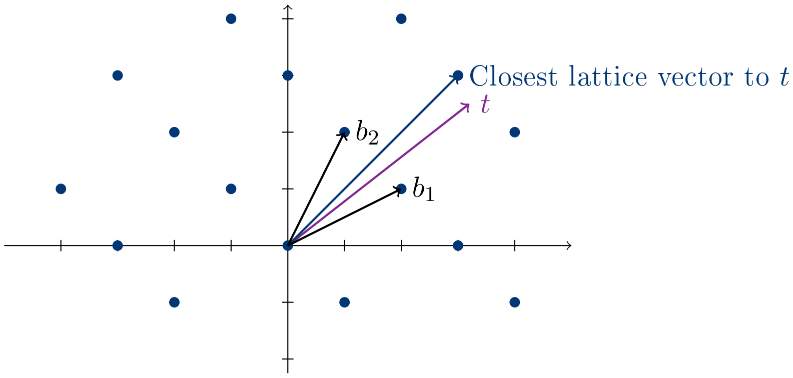

Figure 4.1.: **The closest vector problem.** We consider the lattice generated by the basis vectors $b_1 = (2,1)^T \in \mathbb{R}^2$ and $b_2 = (1,2)^T \in \mathbb{R}^2$. The closest lattice vector to the target vector $t = (3.2, 2.5)^T \in \mathbb{R}^2$ with respect to the Euclidean norm is the vector $b_1 + b_2 = (3,3)^T \in \mathbb{R}^2$.

Definition 4.1.5. *(γ-Approximate Successive Minima Problem (*SMP$_\gamma$*))*
Given a lattice $L \subseteq \mathbb{R}^n$ of rank m, find m linearly independent vectors $v_1, \dots, v_m \in L$ such that

$$||v_i|| \leq \gamma \cdot \lambda_i^{(||\cdot||)}(L)$$

for all $i = 1, \dots, m$.

Definition 4.1.6. *(γ-Approximate Shortest Independent Vectors Problem (*SIVP$_\gamma$*))*
Given a lattice $L \subseteq \mathbb{R}^n$ of rank m, find m linearly independent vectors $v_1, \dots, v_m \in L$ such that

$$||v_i|| \leq \gamma \cdot \lambda_m^{(||\cdot||)}(L)$$

for all $i = 1, \dots, m$.

Another important lattice problem is the closest vector problem which is a (somewhat) inhomogeneous variant of the shortest vector problem. Here, we are given the lattice together with a target vector from the vector space spanned by the lattice. The goal is to find a lattice vector with minimal distance to this target vector. The closest vector problem is illustrated in Figure 4.1.

Definition 4.1.7. *(Closest Vector Problem (*CVP*))*
Given a lattice $L \subseteq \mathbb{R}^n$ and some target vector $t \in \operatorname{span}(L)$, find a lattice vector $v \in L$ such that

$$||v - t|| \leq \min\left\{\|w - t\| \mid w \in L\right\}.$$

We denote by $\mu^{(||\cdot||)}(t, L) := \min\{\|w - t\| \mid w \in L\}$ the minimal distance between the target vector t and the lattice L.

From this point of view, the covering radius of a lattice L is the smallest radius ρ such that for any target vector $t \in \operatorname{span}(L)$ there exists a lattice vector within distance of at most ρ, i.e.,

$$\mu^{(\|\cdot\|)}(L) = \max\left\{\mu^{(\|\cdot\|)}(t, L) \mid t \in \operatorname{span}(L)\right\}.$$

As for the other lattice problems we can define a decisional and an optimization variant of the closest vector problem. For all ℓ_p-norms with $1 \leq p \leq \infty$ and all polyhedral norms these variants are equivalent, see [MG02] and [BN11].

Definition 4.1.8. *(γ-approximate closest vector problem (CVP_γ))*
Given a lattice $L \subseteq \mathbb{R}^n$ and some target vector $t \in \operatorname{span}(L)$, find a lattice vector $v \in L$ such that

$$\|v - t\| \leq \gamma \cdot \mu^{(\|\cdot\|)}(t, L).$$

If we consider these lattice problems with respect to an ℓ_p-norm, we will denote this by $\mathrm{SVP}^{(p)}$, $\mathrm{SMP}^{(p)}$, $\mathrm{SIVP}^{(p)}$, or $\mathrm{CVP}^{(p)}$ respectively. If the norm is described as the Minkowski function of a convex body $\mathcal{C}$ symmetric about the origin, we denote the corresponding version by $\mathrm{SVP}^{(\mathcal{C})}$, $\mathrm{SMP}^{(\mathcal{C})}$, $\mathrm{SIVP}^{(\mathcal{C})}$, or $\mathrm{CVP}^{(\mathcal{C})}$.

To obtain computational statement for the four lattice problems SVP, SMP, SIVP, and CVP, we always assume $L \subseteq \mathbb{Q}^n$. The *size of a lattice* $L \subseteq \mathbb{Q}^n$ with respect to a basis B is the maximum of the dimension n, the rank m, and the size of the numerators and denominators of the coordinates of the basis vectors. In the sequel, if we speak of the size of a lattice L without referring to some specific basis, we implicitly assume that some basis $[b_1, \ldots, b_m]$ for the lattice L is given.

Let us briefly review the main known hardness results for these four lattice problems. All known hardness results for them hold for the decisional variants of the corresponding problems, whereas all algorithms solve or approximate the search versions of the problems.

It is not hard to see that the decisional variants of SVP, SMP, SIVP, and CVP are in NP. In 1981, van Emde Boas proved that closest vector problem is NP-hard with respect to any ℓ_p-norm with $1 \leq p \leq \infty$, see [vEB81]. Furthermore, he proved that the shortest vector problem is NP-hard with respect to the ℓ_∞-norm. In the same paper, he conjectured that the shortest vector problem with respect to the Euclidean norm is also NP-hard. Solving this task is the big outstanding question in the area of lattice problems. In 1996, Ajtai achieved a remarkable partial success. He showed that the shortest vector problem and the shortest independent vectors problem with respect to any ℓ_p-norm with $1 \leq p \leq \infty$ are NP-hard under randomized reduction, see [Ajt98].

These results have been improved in a long sequence of works. Up to now, we know that for any ℓ_p-norm with $1 \leq p \leq \infty$ the shortest vector problem is NP-hard under randomized reduction, see [Kho05], [Din02], [RR06]. The same results hold for the successive minima problem and the shortest independent vectors problem, see [BS99], [RR06]. The closest vector problem in any ℓ_p-norm with $1 \leq p \leq \infty$ is NP-hard to approximate within some factor $m^{\mathcal{O}(1/\log_2 \log_2 m)}$, where m is the rank of the lattice, see [ABSS93], [DKRS03], and [Din02].

On the other hand, we are able to approximate all these lattice problems using polynomial time approximation algorithms with some approximation factor single exponential in the rank of the lattice. These algorithms go back to an idea of Gauss, [Gau01]. The so-called Gaussian reduction algorithm is a generalization of the Euclidean algorithm to dimension 2 and solves the shortest vector problem with respect to the Euclidean norm exactly. It computes in polynomial time a so-called Gaussian reduced basis and for lattices of rank 2 such a basis always contains a shortest non-zero lattice vector, see for example [MG02]. The Gaussian reduction algorithm can be generalized to arbitrary norms, see [KS96].

It was a breakthrough result when Lenstra, Lenstra, and Lovász presented in the early 1980s a generalization of the Gaussian reduction algorithm to arbitrary dimension. The so-called LLL-algorithm was the first polynomial time algorithm that approximates the shortest vector problem. Although the achieved approximation factor is single exponential in the dimension, the algorithm still has a deep impact in many areas in mathematics and computer science. For more information about the relevance of the LLL-algorithm see [NV10].

The LLL-algorithm is a polynomial time algorithm which computes for a given lattice a so-called LLL-reduced basis.

Definition 4.1.9. *(LLL-reduced basis)*
Let $L \subseteq \mathbb{R}^n$ be a lattice. A basis $B = [b_1, \ldots, b_m]$ is called an LLL-reduced basis of the lattice if $L = \mathcal{L}(B)$ and if B satisfies the following properties:

1. *For all $j < i$ we have*
$$\mu_{i,j} = \frac{\langle b_i, b_j^\dagger \rangle}{\langle b_j^\dagger, b_j^\dagger \rangle} \text{ with } |\mu_{i,j}| \leq \frac{1}{2}.$$

2. *For all $1 \leq i < n$ we have*
$$\frac{3}{4} \cdot \|b_i^\dagger\|_2^2 \leq \|\mu_{i+1,i} b_i^\dagger + b_{i+1}^\dagger\|_2^2.$$

To obtain a trade off between the approximation factor and the running time the notion of an LLL-reduced basis can be parameterized using a parameter δ satisfying $1/4 < \delta < 1$. We neglect this aspect here.

Theorem 4.1.10. *(LLL-algorithm, [LLL82])*
Given a lattice basis $B \in \mathbb{Z}^{m \times n}$, the LLL-algorithm computes an LLL-reduced basis using

$\mathcal{O}(n^5 \cdot \log_2(r))$ arithmetic operations on integers of length at most $\mathcal{O}(n^2 \log_2(r))$, where r is an upper bound on the size of the basis vectors.

A complete description of the LLL-algorithm together with a proof of this result can be found for example in [MG02] or [vzGG03]. In the following theorem, we state the main properties of an LLL-reduced basis.

Theorem 4.1.11. *Let $B = [b_1, \dots, b_m]$ be an LLL-reduced basis of a lattice $L \subseteq \mathbb{R}^n$. Then*

- *$\|b_i\|_2 \leq 2^{(m-1)/2}\lambda_i^{(2)}(L)$ for all $1 \leq i \leq m$ and*
- *for all $1 \leq i < j \leq m$, we have*

$$\|b_i^\dagger\|_2^2 \leq 2^{j-i}\|b_j^\dagger\|_2^2.$$

This shows that the LLL-algorithm can be used to compute a $2^{(m-1)/2}$-approximation of the successive minima of a lattice, in particular of its minimum distance. There exist some (slight) improvements and generalizations of the LLL-algorithm due to Schnorr, see [Sch94]. Furthermore, we observe that the LLL-algorithm can be used to solve the shortest vector problem in fixed dimension exactly in polynomial time.

Lovász and Scarf adapted the LLL-algorithm to arbitrary norms, see [LS92]. This algorithm is called the generalized basis reduction algorithm. Unfortunately, it cannot be guaranteed that the number of arithmetic operations of the generalized basis reduction algorithm is polynomially bounded in the dimension.

For the closest vector problem, there exist two polynomial time algorithms that achieve single exponential approximation factor, see [Bab86]. In 1986, Babai showed that a simple rounding method can be used to obtain a c^m-approximation of the closest vector problem for some fixed constant c: For a given target vector $t \in \mathbb{Q}^n$ from the vector space spanned by the lattice, we consider the representation of t as a linear combination of the basis vectors of some LLL-reduced basis of the lattice, $t = \sum_{i=1}^m t_i b_i$ for $t_i \in \mathbb{Q}$, $1 \leq i \leq m$. Babai showed that the lattice vector $\sum_{i=1}^m \lfloor t_i \rceil b_i$ is a c^m approximation of the closest lattice vector to t with respect to the Euclidean norm.
Furthermore, he presented in his paper a variant of the LLL-algorithm that can be used to obtain a polynomial time approximation algorithm for the closest vector problem. The achieved approximation factor is also single exponential in the rank of the lattice.

Theorem 4.1.12. *(Nearest-plane-algorithm, [Bab86])*
Given a lattice $L \subseteq \mathbb{Z}^n$ of rank m and some target vector $t \in \mathbb{Z}^n \cap \operatorname{span}(L)$, the nearest-plane-algorithm computes in polynomial time a vector $v \in L$ such that

$$\|v - t\|_2 \leq 2^{m/2}\mu^{(2)}(t, L).$$

Again, there exists some improvements of this algorithm, see [Sch87], [Kan87a], and [Sch94].

Of course, all these polynomial time approximation algorithms can be generalized to arbitrary ℓ_p-norms with $1 \leq p \leq \infty$ using Hölder's inequality. In this case, the approximation factor increases by the factor roughly $\sqrt{n}$. The same holds also for all tractable norms: If $\|\cdot\|$ is a tractable norm on $\mathbb{R}^n$ where $c \in \mathbb{Z}[X]$ is a polynomial such that $2^{-c(n)}\|x\|_2 \leq \|x\| \leq 2^{c(n)}\|x\|_2$ for all $x \in \mathbb{R}^n$, the LLL-algorithm or the nearest-plane-algorithm can be used to solve Svp or Cvp with respect to the norm $\|\cdot\|$ with approximation factor $2^{c(n)}2^{m/2}$, where m is the rank of the lattice.

Between these two extremes, the NP-hardness of the lattice problems with small approximation factors and the existence of polynomial time algorithms which achieve single exponential time approximation factors, there is a wide gap. Over the last years, a great effort by researchers was spent to close this gap. For example, one can show that approximating Svp or Cvp with respect to an arbitrary ℓ_p-norm, $1 \leq p \leq \infty$, with an almost linear factor is NP-hard unless P = NP, see [LLS90], [Hås88], [Ban93]. For a nice survey on these results see [Reg10] and [Kho10].

In the rest of this thesis, we concentrate on positive results, i.e., known algorithms that solve the lattice problems Svp, Smp, Sivp, and Cvp (almost) optimally. As we have seen, we cannot expect to obtain polynomial time algorithms. Before we focus on algorithms that solve the lattice problems with respect to arbitrary norms, we shortly review the main algorithms that solve the lattice problems with respect to the Euclidean norm and discuss whether they can be adapted to arbitrary norms.

In a breakthrough paper, Micciancio and Voulgaris describe a deterministic single exponential time algorithm that solves the closest vector problem with respect to the Euclidean norm exactly, see [MV10a]. It is based on the computation of the Voronoi cell of a lattice. Using this algorithm, we also obtain a deterministic single exponential time algorithm for the other lattice problems.

Theorem 4.1.13. *(Voronoi-based algorithms for* Svp$^{(2)}$*,* Smp$^{(2)}$*,* Sivp$^{(2)}$*, and* Cvp$^{(2)}$*, [MV10a])*
There exist deterministic algorithms that solve Svp$^{(2)}$*,* Smp$^{(2)}$*,* Sivp$^{(2)}$*, and* Cvp$^{(2)}$*. The number of arithmetic operations of these algorithms is* $2^{(2+o(1))n}\log_2(r)^{\mathcal{O}(1)}$ *and each number computed by the algorithm has bit size* $\log_2(r)^{\mathcal{O}(1)}$*, where n is the rank of the lattice and r is an upper bound on the size of the basis defining the lattice. The space used by the algorithms is* $2^{(1+o(1))n}\log_2(r)^{\mathcal{O}(1)}$.

The algorithms can be easily generalized to all norms which are generated by an inner product or equivalently to general Euclidean norms as remarked in [DPV11]. Unfortunately, it seems that the Voronoi-based algorithms cannot be generalized to other norms since then the Voronoi cell of a lattice is not necessarily convex.

The disadvantage of the algorithms of Miccancio and Voulgaris is that they use exponential space.

The fastest algorithm for the shortest vector problem which uses polynomial space is an algorithm due to Kannan invented in 1983 and refined in 1985 by Helfrich. Recently, the analysis of the algorithm was improved by Hanrot and Stehlé, see [Kan87b], [Hel85], and [HS07].

Theorem 4.1.14. *(Kannan's algorithm for* SVP$^{(2)}$*, [Kan87b], [Hel85], [HS07])*
There exists a deterministic polynomially space bounded algorithm that solves SVP *with respect to the Euclidean norm. The number of arithmetic operations of the algorithm is* $2^{\mathcal{O}(n)} n^{n/(2e)} \log_2(r)^{\mathcal{O}(1)}$ *and each number computed by the algorithm has bit size of at most* $(\log_2(r))^{\mathcal{O}(1)}$*, where n is the rank of the lattice, r is an upper bound on the size of the basis defining the lattice, and e is Euler's constant.*

This algorithm can be adapted easily to arbitrary ℓ_p-norms. In this case, the number of arithmetic operations is $2^{\mathcal{O}(n)} n^n \log_2(r)^{\mathcal{O}(1)}$, see [Kan87b].

For the closest vector problem with respect to the Euclidean norm, there exist basically two algorithms that run in polynomial space. Since there exist polynomial rank preserving reductions from the successive minima problem and the shortest independent vectors problem to the closest vector problem, these polynomially space bounded algorithms also solve the successive minima problem and the shortest independent vectors problem with respect to the Euclidean norm.

One of the polynomially space bounded algorithms for CVP is also due to Kannan and is improved by Helfrich and Hanrot and Stehlé.

Theorem 4.1.15. *(Kannan's algorithm for* CVP$^{(2)}$*, [Kan87b], [Hel85], [HS07])*
There exists a deterministic polynomially space bounded algorithm that solves the closest vector problem with respect to the Euclidean norm. The number of arithmetic operations of the algorithm is $2^{\mathcal{O}(n)} n^{n/2} \log_2(r)^{\mathcal{O}(1)}$ *and each number computed by the algorithm has bit size of at most* $(\log_2(r))^{\mathcal{O}(1)}$*, where n is the rank of the lattice and r is an upper bound on the size of the basis defining the lattice.*

Another algorithm that solves CVP optimally is due to Blömer, see [Blö00].

Theorem 4.1.16. *(Algorithm for* CVP$^{(2)}$ *based on dual* HKZ*-bases, [Blö00])*
There exists a deterministic polynomially space bounded algorithm that solves the closest vector problem with respect to the Euclidean norm. The number of arithmetic operations of the algorithm is $2^{\mathcal{O}(n)} n! \log_2(r)^{\mathcal{O}(1)}$ *and each number computed by the algorithm has bit size of at most* $(\log_2(r))^{\mathcal{O}(1)}$*, where n is the rank of the lattice and r is an upper bound on the size of the basis defining the lattice.*

It may be difficult to generalize these two algorithms to non-Euclidean norms (although Kannan claims the opposite in his paper), since they both use orthogonal projections: At some stage during the algorithm, they consider a target vector which is not contained in the vector space spanned by the lattice. Since it is not possible in this situation to give an upper bound on the distance between the target vector and the lattice, they consider the orthogonal projection of the target vector onto the subspace spanned by the lattice. Unfortunately, if we consider non-Euclidean norms as for example arbitrary ℓ_p-norms, then the closest lattice vector to the target vector is not a closest lattice vector to the orthogonal projection of the target vector or vice versa. Also, if we use norm projections as defined in [Man99] or [LS92], this is not true. We will focus on this aspect in the next Section.

4.2. Similarities and differences of the lattice problems

We now consider the four lattice problems Svp, Smp, Sivp, and Cvp in detail. First of all, we bring up an aspect that we already mentioned in the last section, orthogonal projections. This technique is used in all deterministic algorithms that solve the closest vector problem with respect to the Euclidean norm. We give examples why it is difficult to generalize this technique to norms which are not based on an inner product.

4.2.1. Orthogonal Projections

As we have seen in Chapter 2, we can distinguish between two types of norms on the vector space $\mathbb{R}^n$: The norms which are induced by an inner product and the norms which are not. To recall, all general Euclidean norms are induced by an inner product, whereas all ℓ_p-norms with $p \neq 2$ are not induced by an inner product. The general Euclidean norms on $\mathbb{R}^n$ are exactly the norms whose unit ball is an ellipsoid. For such norms the solution of the closest vector problem can be easily reduced to the solution of the closest vector problem with respect to the Euclidean norm using the fact that each ellipsoid is the image of the Euclidean unit ball under a bijective affine transformation, see Lemma 2.2.7 in Chapter 2.

As we already mentioned, the technique of orthogonal projections plays an important roll in the algorithmic treatment of lattice problems, e.g. it is used in the Cvp$^{(2)}$-algorithms of Kannan and Blömer. In this section, we will show why it does not seem possible to use projections for algorithmic solution of the closest vector problem for norms which are not based on an inner product. We start with a description of the situation and show how we can use projections if we consider the closest lattice vector problem with respect to a norm induced by an inner product. Then, we give an example why this does not seem to work for norms which are not induced by an inner product.

In the following, we assume that we are given a lattice $L = \mathcal{L}(b_1, \dots, b_{n-1}) \subseteq \mathbb{R}^n$ by the basis vectors $b_1, \dots, b_{n-1} \in \mathbb{R}^n$ together with some target vector $t \in \mathbb{R}^n$. Let $b_n \in \mathbb{R}^n$ be a vector such that $[b_1, \dots, b_n]$ is a basis of the vector space $\mathbb{R}^n$. We are searching for the lattice vector in L which is closest to this target vector t. This situation is illustrated

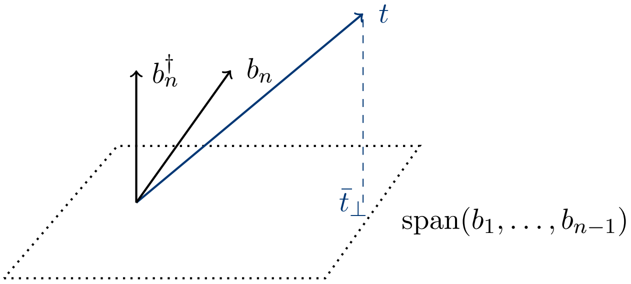

Figure 4.2.: **Projection in a subspace.** The vector t lies in $\text{span}(b_1, \ldots, b_n)$. The vector $\bar{t}_\perp$ denotes the orthogonal projection of t onto $\text{span}(b_1, \ldots, b_{n-1})$.

in Figure 4.2.

Since $t \notin \text{span}(L)$, the distance between the target vector and the lattice can be arbitrarily large. In order to handle this problem we consider the orthogonal projection of t in $\text{span}(b_1, \ldots, b_{n-1})$, which is given by

$$\bar{t}_\perp = t - \pi_n(t) = t - \frac{\langle t, b_n^\dagger \rangle}{\langle b_n^\dagger, b_n^\dagger \rangle} b_n^\dagger, \tag{4.1}$$

where $b_n^\dagger$ is the n-th Gram-Schmidt-vector of the basis $[b_1, \ldots, b_n]$, see Section 3.1 in Chapter 3. If we are searching for a solution of the closest vector problem with respect to the Euclidean norm, we can show the following:

Proposition 4.2.1. *Let $L = \mathcal{L}(b_1, \ldots, b_{n-1}) \subseteq \mathbb{R}^n$ be a lattice and $t \in \mathbb{R}^n$ some target vector. The vector $v \in L$ is a closest lattice vector to t with respect to the Euclidean norm if and only if v is a lattice vector in L closest to the projection $\bar{t}_\perp$ of t in* $\text{span}(b_1, \ldots, b_{n-1})$.

Proof. Let $y \in L \subseteq \text{span}(b_1, \ldots, b_{n-1})$ be a closest lattice vector to t. Since the Euclidean norm is induced by an inner product, we have $\|t - y\|_2^2 = \langle t - y, t - y \rangle$. Using that $t = \bar{t}_\perp + \left(\langle t, b_n^\dagger \rangle / \langle b_n^\dagger, b_n^\dagger \rangle \right) b_n^\dagger$, see Equation (4.1), we obtain

$$\|t - y\|_2^2 = \langle \bar{t}_\perp - y, \bar{t}_\perp - y \rangle + 2 \frac{\langle t, b_n^\dagger \rangle}{\langle b_n^\dagger, b_n^\dagger \rangle} \left\langle b_n^\dagger, \bar{t}_\perp - y \right\rangle + \left\langle \frac{\langle t, b_n^\dagger \rangle}{\langle b_n^\dagger, b_n^\dagger \rangle} b_n^\dagger, \frac{\langle t, b_n^\dagger \rangle}{\langle b_n^\dagger, b_n^\dagger \rangle} b_n^\dagger \right\rangle .$$

Since $b_n^\dagger$ is orthogonal to $\bar{t}_\perp - y \in \text{span}(L)$, we get

$$\|t - y\|_2^2 = \|\bar{t}_\perp - y\|_2^2 + \| \frac{\langle t, b_n^\dagger \rangle}{\langle b_n^\dagger, b_n^\dagger \rangle} b_n^\dagger \|_2^2,$$

where the term $\| \left(\langle t, b_n^\dagger \rangle / \langle b_n^\dagger, b_n^\dagger \rangle \right) b_n^\dagger \|_2^2$ is independent of the lattice vector y. Hence, we see that $\|t - v\|_2$ is minimized over L if and only if $\|\bar{t}_\perp - v\|_2$ is minimized over L. □

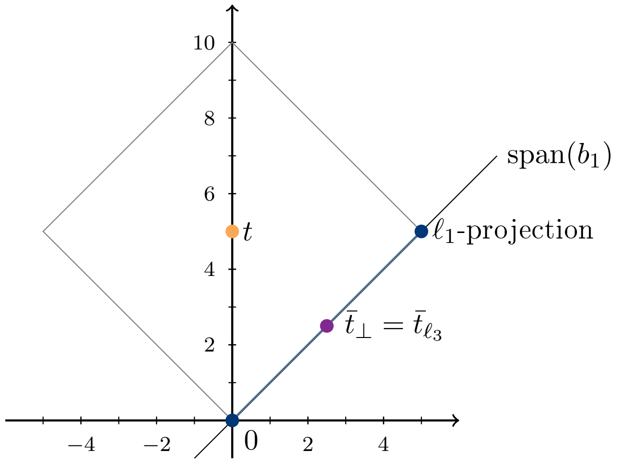

Figure 4.3.: **Norm projections.** We consider the vector space spanned by the vector $b_1 = (1,1)^T \in \mathbb{R}^2$ together with the target vector $t = (0,5)^T \in \mathbb{R}^2$. The orthogonal projection of t in span(b_1) is $\bar{t}_\perp = (2.5, 2.5)^T \in \mathbb{R}^2$, which is also the ℓ_3-projection. The ℓ_1-projection is the whole segment $k \cdot b_1$, $0 \leq k \leq 5$.

This result can be easily adapted to all norms which are based on an inner product, i.e., if the norm $\|\cdot\|$ is defined by $\|x\| = \sqrt{s(x,x)}$, for $x \in \mathbb{R}^n$, where $s : \mathbb{R}^{n \times n} \to \mathbb{R}$ is a symmetric bilinear mapping satisfying the corresponding properties, see Definition 2.2.1 in Chapter 2. Then we use instead of the orthogonal projection defined in (4.1), the projection $t - \left(s(t, b_n^\dagger)/s(b_n^\dagger, b_n^\dagger)\right) b_n^\dagger$, where $b_n^\dagger$ is a vector orthogonal to span$(b_1, \ldots, b_{n-1})$, with respect to the inner product defined by s, i.e., $s(b_n^\dagger, b_i) = 0$ for all $1 \leq i \leq n-1$.

In the rest of this section, we show that Proposition 4.2.1 is not true if the norm is not induced by an inner product. Additionally, we show that this statement is not true if we consider the corresponding norm projection instead of the orthogonal projection: As the norm projection of a vector in a subspace we understand the vector in the subspace with minimal distance in the corresponding norm. That means, we consider

$$\bar{t}_{\text{min}} \in \text{span}(L) \text{ with } \|t - \bar{t}_{\text{min}}\| = \min\left\{\|t - \bar{x}\| \mid \bar{x} \in \text{span}(L)\right\}. \tag{4.2}$$

Mangasarian gave an explicit closed form for this projection using the dual norm, see [Man99]. If we consider a norm induced by an inner product, the norm projection and the orthogonal projection coincide. Additionally, we need to observe that the norm projection might not be uniquely determined if the norm is not strictly convex. In Figure 4.3 we see an example of different norm projections.

In the following, we give examples which show that Proposition 4.2.1 does not hold for non-Euclidean norms. We consider two norms in detail. First, we consider the ℓ_1-norm which is very descriptive. Then we consider a strictly convex norm, the ℓ_3-norm.

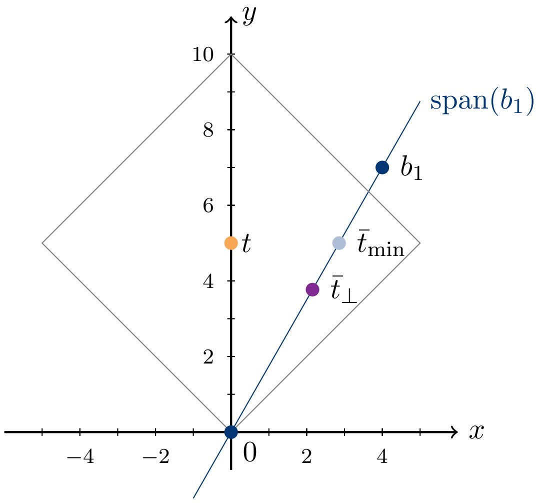

Figure 4.4.: **Counterexample for projections with respect to the ℓ_1-norm.** We consider the lattice spanned by the vector b_1 together with the target vector t. The vector $\bar{t}_\perp$ is the orthogonal projection of t in $\operatorname{span}(b_1)$, $\bar{t}_{\min}$ is the ℓ_1-projection.

Projection with respect to the ℓ_1-norm We consider the vector space $\mathbb{R}^2$ and the lattice spanned by the vector $b_1 = (4,7)^T \in \mathbb{R}^2$. Additionally, we consider the target vector $t = (0,5)^T \in \mathbb{R}^2$ which is not contained in the subspace $\operatorname{span}(b_1)$. We are searching for a lattice vector in $\mathcal{L}(b_1)$ which is closest to t with respect to the ℓ_1-norm, see Figure 4.4 for an illustration.

Claim 4.2.2. *The vector $v = 0$ is the closest lattice vector to t in $\mathcal{L}(b_1)$ with respect to the ℓ_1-norm.*

Proof. Every lattice vector $v \in \mathcal{L}(b_1)$ is of the form $v = v_1 b_1 = (4v_1, 7v_1)^T$ with $v_1 \in \mathbb{Z}$. Using this representation, the distance between t and a lattice vector is given by $\|t - v_1 b_1\|_1 = 4|v_1| + |5 - 7v_1|$ and it becomes minimal over $\mathbb{Z}$ if $v_1 = 0$. □

Now, we consider the orthogonal projection $\bar{t}_\perp$ of t in $\operatorname{span}(b_1)$ with respect to the Euclidean norm, see Equation (4.1). The vector $(-7,4)^T$ is orthogonal to b_1. Hence, $\bar{t}_\perp$ is given by

$$\bar{t}_\perp = t - \frac{\left\langle t, \begin{pmatrix} -7 \\ 4 \end{pmatrix} \right\rangle}{\left\langle \begin{pmatrix} -7 \\ 4 \end{pmatrix}, \begin{pmatrix} -7 \\ 4 \end{pmatrix} \right\rangle} \begin{pmatrix} -7 \\ 4 \end{pmatrix} = \frac{7}{13} \begin{pmatrix} 4 \\ 7 \end{pmatrix}.$$

We are searching for the closest lattice vector to $\bar{t}_\perp$ with respect to the ℓ_1-norm. Obviously, in a lattice of rank 1 we get the closest lattice vector by rounding.

Claim 4.2.3. *The vector b_1 is a closest lattice vector to $\bar{t}_\perp = (7/13) \cdot (4,7)^T$ in $\mathcal{L}(b_1)$ with respect to the ℓ_1-norm.*

Hence, this is an example where the lattice vector which is closest to t is not the lattice vector which is closest to the orthogonal projection of t in the lattice. We now consider the vector $\bar{t}_{\min} \in \text{span}(b_1)$ which is closest to t with respect to the ℓ_1-norm, as defined in (4.2).

The ℓ_1-projection of a vector t onto a subspace S depends of the orientation of the subspace. In $\mathbb{R}^2$, if the angle θ is different from $\pi/4$, the projection is unique but directly along the y-axis or the x-axis. If $\theta = \pi/4$, the projection is a segment and it includes the points along both unit directions.
In our example, we obtain

$$\begin{aligned} \min\left\{\|t - \bar{x}\|_1 \mid \bar{x} \in \text{span}(b_1)\right\} &= \min\left\{\left\| \begin{pmatrix} 0 \\ 5 \end{pmatrix} - x_1 \begin{pmatrix} 4 \\ 7 \end{pmatrix} \right\|_1 \mid x_1 \in \mathbb{R}\right\} \\ &= \min\left\{4|x_1| + |5 - 7x_1| \mid x_1 \in \mathbb{R}\right\} \end{aligned}$$

This value becomes minimal if $x_1 = 5/7$. Hence, $\bar{t}_{\min} = (5/7) \cdot (4,7)^T$. Obviously, we get the following result.

Claim 4.2.4. *The vector $b_1 = (4,7)^T$ is the closest lattice vector to $\bar{t}_{\min}$ in $\mathcal{L}(b_1)$ with respect to the ℓ_1-norm.*

Hence, this is additionally an instance of the closest vector problem where a lattice vector that is closest to t is not closest to the target vector $\bar{t}_{\min}$ which is the ℓ_1-projection of t in span(L).

Projection with respect to the ℓ_3-norm We consider the $\mathbb{R}^2$ and the lattice spanned by the vector $b_1 = (-12, 44)^T \in \mathbb{R}^2$. Additionally, we consider the target vector $t = (-20, 19)^T \in \mathbb{R}^2$, which is not contained in the subspace span(b_1). We are searching for a lattice vector in $\mathcal{L}(b_1)$ which is closest to t with respect to the ℓ_3-norm.

Claim 4.2.5. *The vector $v = 0$ is the closest lattice vector to t in $\mathcal{L}(b_1)$ with respect to the ℓ_3-norm.*

Proof. Every lattice vector $v \in \mathcal{L}(b_1)$ is of the form $v = v_1 b_1 = (-12v_1, 44v_2)^T$ with $v_1 \in \mathbb{Z}$. Using this representation, the distance between t and a lattice vector in $\mathcal{L}(b_1)$ becomes minimal over $\mathbb{Z}$ if

$$\|t - v_1 b_1\|_3^3 = |-20 + 12v_1|^3 + |19 - 44v_1|^3$$

becomes minimal over $\mathbb{Z}$, i.e., if $v_1 = 0$. □

We now consider the orthogonal projection $\bar{t}_\perp$ of t in span(b_1) with respect to the Euclidean norm. The vector $(44, 12)^T$ is orthogonal to b_1. Hence, $\bar{t}_\perp$ is given by

$$\bar{t}_\perp = t - \frac{\left\langle t, \begin{pmatrix} 44 \\ 12 \end{pmatrix} \right\rangle}{\left\langle \begin{pmatrix} 44 \\ 12 \end{pmatrix}, \begin{pmatrix} 44 \\ 12 \end{pmatrix} \right\rangle} \begin{pmatrix} 44 \\ 12 \end{pmatrix} = \frac{269}{520} \begin{pmatrix} -12 \\ 44 \end{pmatrix}$$

Since $\mathcal{L}(b_1)$ is a lattice of rank 1, we obtain the closest lattice vector to $\bar{t}_\perp$ with respect to the ℓ_3-norm by rounding.

Claim 4.2.6. *The vector b_1 is a closest lattice vector to $\bar{t}_\perp$ in $\mathcal{L}(b_1)$ with respect to the ℓ_3-norm.*

To compute the vector $\bar{t}_{\min} \in$ span(b_1) which is the closest vector to t in span(b_1), we are searching for

$$\min \left\{ \|t - \lambda b_1\|_3^3 \mid \lambda \in \mathbb{R} \right\} = \min \left\{ |-20 + 12\lambda|^3 + |19 - 44\lambda|^3 \mid \lambda \in \mathbb{R} \right\},$$

see Equation (4.2). Using standard techniques, it is easy to compute that this minimum is achieved for $\lambda = (13 + \sqrt{33})/32$, i.e., $\bar{t}_{\min} = ((13 + \sqrt{33})/32) \cdot b_1$. Hence we obtain the following result.

Claim 4.2.7. *The vector $b_1 = (-12, 44)^T$ is the closest lattice vector to $\bar{t}_{\min}$ in $\mathcal{L}(b_1)$ with respect to the ℓ_3-norm.*

These examples illustrate why it is not possible to use orthogonal projections for the solution of the closest vector problem with respect to non-Euclidean norms.

4.2.2. Number of solutions

In this section, we study the question if there exists an upper bound on the number of optimal solutions for lattice problems. We will show that the shortest vector problem is the only lattice problem among the four classical lattice problems where the number of almost optimal solutions is at most single exponential in the dimension. This result holds for any norm.

The results presented in this section are based on results of Niemeier for the number of optimal solutions of the shortest vector problem and the closest vector problem for arbitrary ℓ_p-norms with $1 < p < \infty$, see [Nie07].

First of all, we show that for every strictly convex norm the number of exact solutions of Svp, Smp, and Cvp is at most 2^{m+1} where m is the rank of the lattice.

To show this, we consider the cosets of the group $L/2L$. To recall, two lattice vectors $v, w \in L$ are contained in the same coset if and only if $v - w \in 2L$, i.e, if $(v - w)/2 \in L$, see Definition 3.1.5 in Chapter 3. The number of cosets of $L/2L$ is exactly 2^m. Given a basis $B = [b_1, \ldots, b_m]$ of the lattice L every lattice vector $v \in L$ can be uniquely represented as $v = \sum_{i=1}^m (2\bar{v}_i + \hat{v}_i) b_i$ with $\bar{v}_i \in \mathbb{Z}$ and $\hat{v}_i \in \{0, 1\}$ for all $1 \leq i \leq m$. Using this representation, two lattice vectors $v, w \in L$ are contained in the same coset if and only if $\hat{v}_i = \hat{w}_i$ for all $1 \leq i \leq m$.

The main idea to give an upper bound on the number of solutions is as follows: If two lattice vectors $v, w \in L$ are contained in the same coset, the vector $(v+w)/2$ is also contained in the lattice L and it follows from the strict convexity of the norm that

$$\|\frac{w+v}{2}\| < \frac{1}{2}\|w\| + \frac{1}{2}\|v\|.$$

That means if v and w have the same length, the vector $(v+w)/2$ is a shorter lattice vector than v and w. This leads directly to an upper bound on the number of possible solutions for the shortest vector problem.

Lemma 4.2.8. *Let $\|\cdot\|$ be a strictly convex norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m. Then the number of shortest non-zero lattice vectors in L with respect to the norm $\|\cdot\|$ is at most 2^{m+1}.*

Proof. Let $u, v \in L \backslash \{0\}$, be two shortest distinct lattice vectors, that means

$$\|u\| = \|v\| = \lambda_1^{(\|\cdot\|)}(L) \text{ and } u \neq \pm v.$$

If u and v are contained in the same coset $L/2L$, then $(v-w)/2 \in L$. Since L is an additive subgroup of $\mathbb{R}^n$, this shows that also $(1/2)(v-w)+w = (1/2)(v+w) \in L\backslash\{0\}$. Since the norm is strictly convex, we have

$$\|\frac{1}{2}(u+v)\| < \frac{1}{2}\|u\| + \frac{1}{2}\|v\| = \lambda_1^{(\|\cdot\|)}(L),$$

which yields a contradiction. Hence, every coset contains at most two shortest non-zero lattice vectors $v_1, v_2 \in L$ which satisfy $v_1 = -v_2$. Since the number of cosets of $L/2L$ is exactly 2^m, the number of shortest non-zero lattice vectors is at most $2 \cdot 2^m = 2^{m+1}$. □

We can use the same argument to give an upper bound on the number of optimal solutions for the successive minima problem.

Lemma 4.2.9. *Let $\|\cdot\|$ be a strictly convex norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m. For $1 < i \leq m$, let $v_1, \ldots, v_{i-1} \in L$ be linearly independent such that $\|v_j\| = \lambda_j^{(\|\cdot\|)}(L)$ for $1 \leq j < i$. Then the number of shortest lattice vector $v \in L$ with $v \notin \operatorname{span}(v_1, \ldots, v_{i-1})$ is at most 2^{m+1}.*

Proof. Let $u, v \in L$ be two distinct vectors satisfying $u, v \notin \operatorname{span}(v_1, \ldots, v_{i-1})$, $u \neq \pm v$, and $\|u\| = \|v\| = \lambda_j^{(\|\cdot\|)}(L)$. If u and v are contained in the same coset, we have $(u-v)/2 \in L$. Since u and v are not contained in $\operatorname{span}(v_1, \ldots, v_{i-1})$, we have either $(u+v)/2 \notin \operatorname{span}(v_1, \ldots, v_{j-1})$ or $(u-v)/2 \notin \operatorname{span}(v_1, \ldots, v_{j-1})$. Without loss of generality we assume that $(u+v)/2 \notin \operatorname{span}(v_1, \ldots, v_{j-1})$.

Since the norm is strictly convex, we obtain the contradiction to the definition of the j-th successive minimum that

$$\left\|\frac{1}{2}(u+v)\right\| < \frac{1}{2}\|u\| + \frac{1}{2}\|v\| = \lambda_j^{(\|\cdot\|)}(L).$$

This shows that the vectors u, v are not contained in the same coset. Since the number of cosets of $L/2L$ is exactly 2^m, the number of lattice vectors in $L \setminus \operatorname{span}(v_1, \ldots, v_{j-1})$ with minimal length is at most $2 \cdot 2^m = 2^{m+1}$. □

Also for the closest vector problem we can show that the number of optimal solutions is single exponential in the rank of the lattice.

Lemma 4.2.10. *Let $\|\cdot\|$ be a strictly convex norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice of rank m and $t \in \mathbb{R}^n$ be some target vector. Then the number of lattice vectors in L which are closest to t with respect to the norm $\|\cdot\|$ is at most 2^m.*

Proof. Let $u, v \in L$, $u \neq v$, be closest lattice vectors to t with respect to the norm $\|\cdot\|$. If $u \equiv v \mod 2L$, we have $(1/2)(u-v) \in L$ Since L is an additive subgroup of $\mathbb{R}^n$, it follows that $(u+v)/2 = (u-v)/2 + v \in L$. Using the strict convexity of the norm, we obtain that

$$\left\|\frac{1}{2}(u+v) - t\right\| = \frac{1}{2}\|u+v-2t\| < \frac{1}{2}\|u-t\| + \frac{1}{2}\|v-t\|.$$

Since $u, v \in L$ are closest lattice vectors to t we obtain the contradiction that

$$\left\|\frac{1}{2}(u+v) - t\right\| < \mu^{(\|\cdot\|)}(t, L).$$

Hence every coset contains at most one closest lattice vector to t. Since the number of cosets of $L/2L$ is exactly 2^m, the number of closest lattice vectors is at most 2^m. □

For Smp and Cvp these results do not hold if the norm is not strictly convex, as it is illustrated in Figure 4.5.

For the shortest vector problem we can show that for every norm the number of exact solutions is single exponential in the dimension. This result is based on the following lemma which is a generalization of Claim 5 in [Reg04] based on an idea of Goldreich and Goldwasser, see [GG00].

Lemma 4.2.11. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice, $t \in \mathbb{R}^n$ and $R > 0$. Then the number of lattice vectors in the ball $\bar{B}_n^{(\|\cdot\|)}(t, R)$ is at most*

$$|\bar{B}_n^{(\|\cdot\|)}(t, R) \cap L| < \left(\frac{2R + \lambda_1^{(\|\cdot\|)}(L)}{\lambda_1^{(\|\cdot\|)}(L)}\right)^n.$$

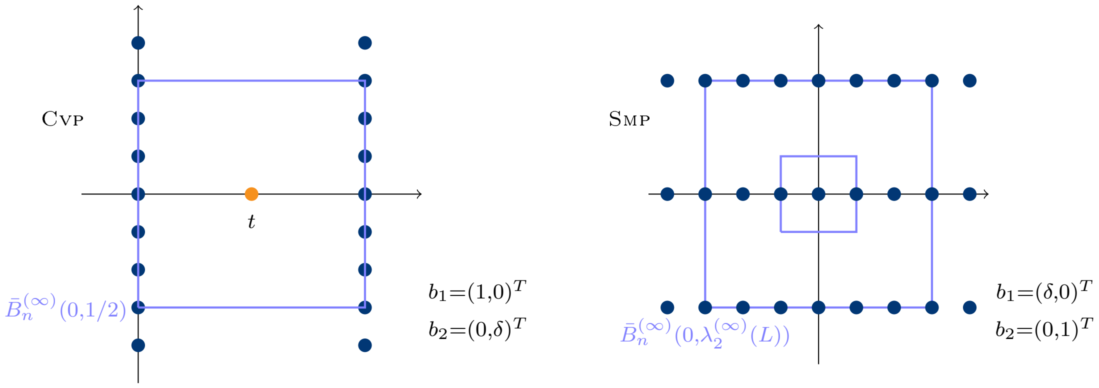

Figure 4.5.: **Number of solutions for Smp and Cvp with respect to not strictly convex norms.** On the left, we consider the lattice generated by the basis vectors $(1,0)^T \in \mathbb{R}^2$ and $(0,\delta)^T \in \mathbb{R}^2$ with $\delta > 0$ small together with the target vector $t = (1/2, 0)$. The distance between this target vector and the lattice in the ℓ_∞-norm is $1/2$. The smaller the parameter δ, the greater the number of lattice vectors in $\bar{B}_n^{(\infty)}(t, 1/2)$. On the right, we consider the lattice generated by the basis vectors $(\delta, 0)^T \in \mathbb{R}^2$ and $(0,1)^T \in \mathbb{R}^2$. The minimum distance of this lattice with respect to the ℓ_∞-norm is δ and the second successive minimum is 1. The smaller the parameter δ is, the greater the number of lattice vectors in $\bar{B}_n^{(\infty)}(0,1)$ which are not contained in $\operatorname{span}((\delta,0)^T)$.

Proof. By definition of the minimum distance, for all lattice vectors $v, w \in L$, $v \neq w$, the balls with radius $\lambda_1^{(\|\cdot\|)}(L)/2$ around these vectors are disjoint,

$$B_n^{(\|\cdot\|)}\left(v, \frac{\lambda_1^{(\|\cdot\|)}(L)}{2}\right) \cap B_n^{(\|\cdot\|)}\left(w, \frac{\lambda_1^{(\|\cdot\|)}(L)}{2}\right) = \emptyset \text{ for all } v, w \in L.$$

If we regard only lattice vectors in $\bar{B}_n^{(\|\cdot\|)}(t, R)$, it follows from the convexity of the norm that their union is contained in $\bar{B}_n^{(\|\cdot\|)}(t, R + \lambda_1^{(\|\cdot\|)}(L)/2)$. Therefore the number of elements in $\bar{B}_n^{(\|\cdot\|)}(t, R) \cap L$ is at most

$$\left|\bar{B}_n^{(\|\cdot\|)}(t, R) \cap L\right| \leq \frac{\operatorname{vol}_n\left(\bar{B}_n^{(\|\cdot\|)}\left(t, R + \frac{\lambda_1^{(\|\cdot\|)}(L)}{2}\right)\right)}{\operatorname{vol}_n\left(B_n^{(\|\cdot\|)}\left(v, \frac{\lambda_1^{(\|\cdot\|)}(L)}{2}\right)\right)} = \left(\frac{2R + \lambda_1^{(\|\cdot\|)}(L)}{\lambda_1^{(\|\cdot\|)}(L)}\right)^n,$$

where the last equality follows from Equation (2.1) in Chapter 2. □

Using this result with radius $R = \gamma \lambda_1^{(\|\cdot\|)}(L)$ for some parameter $\gamma \geq 1$, we obtain the following.

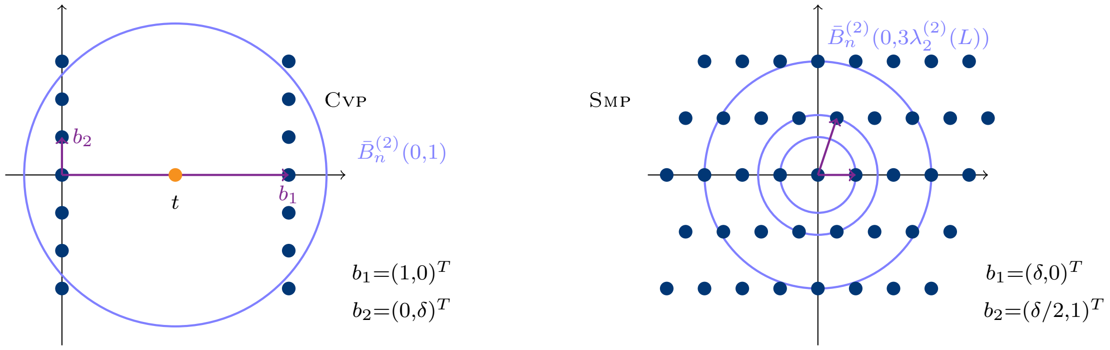

Figure 4.6.: **Number of approximate solutions for** Cvp **and** Smp**.** On the left, we consider the lattice generated by the basis vectors $(1,0)^T \in \mathbb{R}^2$ and $(0,\delta)^T \in \mathbb{R}^2$ with $\delta > 0$ small together with the target vector $t = (1/2,0)^T \in \mathbb{R}^2$. The Euclidean distance between the target vector and the lattice is $1/2$. The smaller the parameter δ, the greater the number of lattice vectors which are contained in Euclidean ball with radius 1 around the target vector t. On the right, we consider the lattice generated by the basis vectors $(\delta,0)^T \in \mathbb{R}^2$ and $(\delta/2,1)^T \in \mathbb{R}^2$. The minimum distance of this lattice in the Euclidean norm is δ, the second successive minimum is $\lambda_2^{(2)}(L) = \sqrt{\delta^2/4+1}$. The smaller the parameter δ is, the greater the number of lattice vectors in the Euclidean ball with radius $3\lambda_2^{(2)}(L)$.

Corollary 4.2.12. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice $t \in \mathbb{R}^n$. Then the number of lattice vectors in L with length at most $\gamma \cdot \lambda_1^{(\|\cdot\|)}(L)$ is at most $(2\gamma+1)^n$, i.e.,*

$$\left| \bar{B}_n^{(\|\cdot\|)}(0, \gamma \cdot \lambda_1^{(\|\cdot\|)}(L)) \cap L \right| \leq (2\gamma+1)^n$$

If we choose $\gamma = 1$, this result shows that the number of optimal solutions of the shortest vector problem for every norm is at most 3^n. Furthermore, it shows that also the number of γ-approximate solutions of Svp is at most single exponential in the dimension. This is the main reason why the solution of the shortest vector problem is comparatively uncomplicated. For the other lattice problems Smp, Sivp, and Cvp, such results do not hold as it is shown in the examples presented in Figure 4.6. The example for Smp also shows why it is not possible to give an upper bound on the number of optimal solutions for Sivp.

Of course, if we consider restricted versions of Smp, Sivp, or Cvp, we can use the result of Lemma 4.2.11. These restricted versions are characterized by the fact that the successive minima respectively the distance of the target vector to the lattice are not much longer than the minimum distance of the lattice.

Corollary 4.2.13. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice satisfying $\lambda_i^{(\|\cdot\|)}(L) \leq c \cdot \lambda_i^{(\|\cdot\|)}(L)$ for some $c > 0$ and some index $1 \leq i \leq n$. Then the number of* *lattice vectors in L with length at most $\gamma \cdot \lambda_i^{(\|\cdot\|)}(L)$ for some $\gamma \geq 1$, is at most $(2\gamma \cdot c+1)^n$, i.e.,*

$$\left|\bar{B}_n^{(\|\cdot\|)}\left(0, \gamma \cdot \lambda_i^{(\|\cdot\|)}(L)\right) \cap L\right| \leq (2\gamma \cdot c+1)^n$$

Corollary 4.2.14. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice and $t \in \operatorname{span}(L)$ be some target vector satisfying $\mu^{(\|\cdot\|)}(t, L) \leq c \cdot \lambda_1^{(\|\cdot\|)}(L)$ for some constant $c > 0$. Then the number of lattice vectors in L whose distance to the target vector is at most $\gamma \cdot \mu^{(\|\cdot\|)}(t, L)$ for some $\gamma \geq 1$, is upper bounded by $(2\gamma \cdot c+1)^n$, i.e.,*

$$\left|\bar{B}_n^{(\|\cdot\|)}\left(0, \gamma \cdot \mu^{(\|\cdot\|)}(t, L)\right) \cap L\right| \leq (2\gamma \cdot c+1)^n.$$

After the general considerations, which make similarities and differences between the lattice problems clear, we now consider the relation between the lattice problems, i.e., we deal with the topic if certain lattice problems are easier than others. Furthermore, we introduce two additional lattice problems that will allow us to present a unified algorithmic treatment of the lattice problems in the following.

4.3. Relation between lattice problems

Up to now we considered the lattice problems relatively independently. Now we want to study the relation between them. Since we focus in this thesis on the complexity of lattice problems with respect to arbitrary norms, we neglect all reductions between lattice problems which work only for the Euclidean norm.

Obviously, there is a polynomial time reduction from the shortest vector problem to the successive minima problem which works for any norm. The same holds for the shortest independent vectors problem and the successive minima problem.

The relation between the shortest vector problem and the closest vector problem is not so obvious. Although, the closest vector problem is considered as a kind of an inhomogeneous version of the shortest vector problem, we have to keep in mind that a shortest vector in some lattice L is not a closest lattice vector to 0 in the lattice L since 0 is always a lattice vector. In 1999, Goldreich, Micciancio, Safra, and Seifert showed that the shortest vector problem is not harder than the closest vector problem, see [GMSS99]. That means there exists a polynomial time reduction from the shortest vector problem to the closest vector problem. This reduction works for any efficiently computable norm and preserves the rank of the lattice. Furthermore, it preserves the approximation factor, i.e., if we are given an algorithm $\mathcal{A}$ that computes for a given lattice $L \subseteq \mathbb{R}^n$ and some target vector $t \in \operatorname{span}(L) \cap \mathbb{R}^n$ a vector $v \in L$ such that $\|v - t\| \leq f(n) \cdot \mu^{(\|\cdot\|)}(t, L)$ for some function $f : \mathbb{N} \to \mathbb{R}^{\geq 0}$, then there exists an algorithm that computes for a given lattice L a vector $v \in L$ satisfying $\|v\| \leq f(n) \cdot \lambda_1^{(\|\cdot\|)}(L)$.

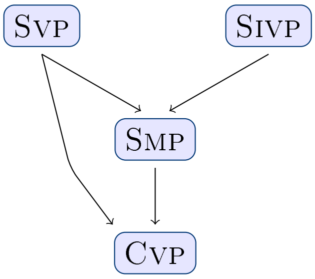

Figure 4.7.: **Relation between the lattice problems for arbitrary norms.** Arrows indicate polynomial time reductions preserving the rank of the lattice and the approximation factor.

In 2008, Micciancio shows the same for the relation between the successive minima problem and the closest vector problem. In [Mic08], he presented a polynomial time reduction form the successive minima problem to the closest vector problem which preserves the rank of the lattice and the approximation factor. The reduction works for any efficiently computable norm.

These relations between the four lattice problems SVP, SMP, SIVP, and CVP are illustrated in Figure 4.7. For the sake of completeness, one can show that with respect to the Euclidean norm, SMP, SIVP, and CVP in their exact version are equivalent and that there exists a polynomial time reduction from the exact version of SVP to all of these problems, see [Mic08].

4.3.1. The generalized shortest vector problem

To obtain a unified algorithmic treatment for all four lattice problems we define a new lattice problem, the generalized shortest vector problem, GSVP. We will show that there are polynomial time reductions from SVP, SMP, SIVP, and CVP to GSVP. In the next chapter, Chapter 5, we will present a probabilistic single exponential time algorithm that approximates the generalized shortest vector problem with approximation factor $1+\epsilon$ for any $0<\epsilon<3/2$.

Definition 4.3.1. *(Generalized Shortest Vector Problem (GSVP))*
Given a lattice $L \subseteq \mathbb{R}^n$ and some subspace $M \subsetneq \operatorname{span}(L)$ find a shortest lattice vector $v \in L \backslash M$ with respect to the norm $\|\cdot\|$. We set

$$\lambda_M^{(\|\cdot\|)}(L) := \min\{r \in \mathbb{R} \mid \exists v \in L\backslash M,\ \|v\| \leq r\}$$

and call it the subspace avoiding minimum*.*

The geometry behind the generalized shortest vector problem is illustrated in Figure 4.8.

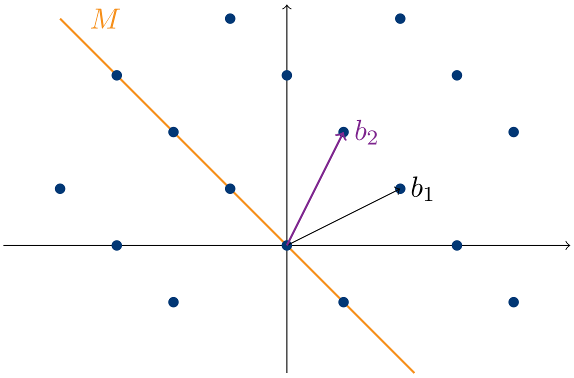

Figure 4.8.: **The generalized shortest vector problem.** The lattice L is generated by the basis vectors $b_1 = (2,1)^T \in \mathbb{R}^2$ and $b_2 = (1,2)^T \in \mathbb{R}^2$, the subspace M is spanned by the vector $v = (-1,1)^T \in \mathbb{R}^2$. A shortest vector in $L \backslash M$ (with respect to the Euclidean norm) is the vector b_2.

We now show that there are polynomial time reductions from Svp, Smp, Sivp, and Cvp to Gsvp as it is illustrated in Figure 4.9. In the following, we are given access to an algorithm $\mathcal{A}$ that solves the generalized shortest vector problem with an approximation factor $1+\epsilon$ for some arbitrary $\epsilon \geq 0$. The core of the reductions is a suitable definition of the subspace.

The shortest vector problem

Theorem 4.3.2. *For all efficiently computable norms the shortest vector problem with approximation factor $1+\epsilon$, $\epsilon \geq 0$, is polynomial time reducible to the generalized shortest vector problem with approximation factor $1+\epsilon$.*

Proof. We choose $M := \{0\} \subsetneq \operatorname{span}(L)$. Hence, if we compute a (almost) shortest lattice vector $u \in L \backslash M$, we compute a (almost) shortest non-zero lattice vector $u \in L$, i.e., we have $\lambda_M^{(\|\cdot\|)}(L) = \lambda_1^{(\|\cdot\|)}(L)$. Therefore, using the algorithm $\mathcal{A}$ with input of the lattice L and the subspace M we get a $(1+\epsilon)$-approximation of a shortest non-zero lattice vector in L. □

The successive minima problem and the shortest independent vectors problem

Theorem 4.3.3. *For all efficiently computable norms the successive minima problem and the shortest independent vectors problem with approximation factor $1+\epsilon$, $\epsilon \geq 0$, are polynomial time reducible to the generalized shortest vector problem with approximation factor $1+\epsilon$.*

Proof. Since Sivp reduces to Smp, we concentrate on the reduction of the successive minima problem to the generalized shortest vector problem. Using the algorithm $\mathcal{A}$, we

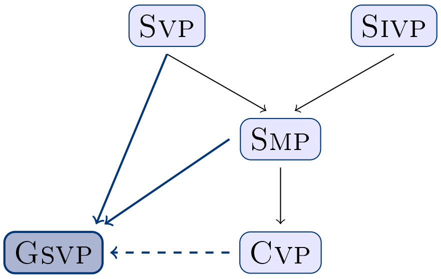

Figure 4.9.: **Relation between GSVP and the other lattice problems.** Arrows indicate polynomial time reductions preserving the rank of the lattice and the approximation factor. The arrow from CVP to GSVP is marked dashed since the approximation factor is not exactly preserved by the reduction.

get a $(1+\epsilon)$-approximation of the first successive minimum as in Theorem 4.3.2. For $i > 1$ we define the subspace $M := \operatorname{span}(v_1, \ldots, v_{i-1})$ with $v_1, \ldots, v_{i-1} \in L$ linearly independent. Since $\dim(M) < i$, there exists a vector $w \in L$ with $\|w\| \leq \lambda_i^{(\|\cdot\|)}(L)$ and $w \notin M$. Therefore, $\lambda_M^{(\|\cdot\|)}(L) < \lambda_i^{(\|\cdot\|)}(L)$ and using the algorithm $\mathcal{A}$ with input of the lattice L and the subspace M we get a $(1+\epsilon)$-approximation for the i-th successive minimum. □

The closest vector problem

The reduction of the closest vector problem to the generalized shortest vector problem relies on a lifting technique introduced by Kannan [Kan87b] and refined by Ajtai, Kumar and Sivakumar [AKS02] and Micciancio and Goldwasser [MG02], respectively.

We assume that we are given an instance of the closest vector problem by a lattice $L \subseteq \mathbb{R}^n$ of rank m and some target vector $t \in \operatorname{span}(L)$. We construct an instance of the generalized shortest vector problem by embedding the lattice and the target vector in a higher dimensional space. We define the $(n+1)$-dimensional lattice L' as the smallest lattice which contains the vector $(t^T, \gamma)^T \in \mathbb{R}^{n+1}$ for some suitable chosen parameter γ and all vectors of the form $(v^T, 0)^T \in \mathbb{R}^{n+1}$ where v is a lattice vector from the original lattice L. If $[b_1, \ldots, b_m] \in \mathbb{R}^{n \times m}$ is a basis of the lattice L, we define

$$L' := \mathcal{L}\left(\left[\begin{pmatrix} b_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} b_m \\ 0 \end{pmatrix}, \begin{pmatrix} t \\ \gamma \end{pmatrix}\right]\right).$$

The parameter $\gamma \in \mathbb{R}$ will be defined later. Additionally, we define the subspace

$$M := \operatorname{span}\left(\left\{\begin{pmatrix} b_i \\ 0 \end{pmatrix} \mid 1 \leq i \leq m\right\}\right) \subseteq \operatorname{span}(L').$$

In the following, we will show that we are able to compute a lattice vector in L which is (almost) closest to the target vector t if we are given a (almost) shortest vector in $L' \backslash M$.

Every vector in $L' \backslash M$ is of the form $(v^T, 0)^T + k(t^T, \gamma)$ with $v \in L$ and $k \in \mathbb{Z}$. If we have $k = -1$, the length of such a vector becomes minimal if and only if the distance between the target vector t and a lattice vector from L becomes minimal.
The main difficulty of the construction is the choice of the parameter γ. We need to choose it appropriately such that a shortest vector in $L' \backslash M$ is of the form described above with $k = -1$.

Another technical difficulty of the construction described above is that we want to solve/approximate the closest vector problem with respect to a tractable norm $\|\cdot\|$ on $\mathbb{R}^n$ using the solution of an instance of the generalized shortest vector problem in $\mathbb{R}^{n+1}$. To do so, we need access to an oracle $\mathcal{A}$ that solves the generalized shortest vector problem with respect to the following norm on $\mathbb{R}^{n+1}$: We define the mapping

$$\begin{aligned} F : \mathbb{R}^{n+1} &\to \mathbb{R} \\ x = (\bar{x}^T, \hat{x})^T &\mapsto \|\bar{x}\| + |\hat{x}|. \end{aligned} \tag{4.3}$$

It is easy to see that F defines a norm on $\mathbb{R}^{n+1}$ if $\|\cdot\|$ defines a norm on $\mathbb{R}^n$. Furthermore, we see that F is a tractable norm if $\|\cdot\|$ is a tractable norm.

Lemma 4.3.4. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. Then the mapping $F : \mathbb{R}^{n+1} \to \mathbb{R}^{\geq 0}$,*

$$\begin{aligned} F : \mathbb{R}^{n+1} &\to \mathbb{R} \\ x = (\bar{x}^T, \hat{x})^T &\mapsto \|\bar{x}\| + |\hat{x}|. \end{aligned}$$

is a tractable norm on $\mathbb{R}^{n+1}$.

In the following, we assume that the oracle $\mathcal{A}$ solves the generalized shortest vector problem with respect to this norm F with an approximation factor $1 + \epsilon$ for any $\epsilon \geq 0$. We show that we are able to solve the closest vector problem with respect to the norm $\|\cdot\|$ exactly if $\mathcal{A}$ solves the generalized shortest vector problem with respect to the norm F exactly. If $\mathcal{A}$ solves the generalized shortest vector problem with respect to the norm F with an approximation factor $1 + \epsilon$, $0 \leq \epsilon \leq 1/2$, we will find a $(1 + 6\epsilon)(1 + \alpha)$-approximation of the closest vector problem with respect to the norm $\|\cdot\|$. Here the parameter $\alpha > 0$ is arbitrary.

The main idea is to try to set the parameter γ to some value slightly bigger than the distance $\mu^{(\|\cdot\|)}(t, L)$ between the target vector and its closest vector in the lattice L. Since we are able to decide in polynomial time whether $t \in L$, we assume $\mu^{(\|\cdot\|)}(t, L) > 0$, see for example [Coh93]. Given a parameter $\alpha > 0$ the reduction requires a parameter $\rho > 0$ with

$$\rho \leq \mu^{(\|\cdot\|)}(t, L) < (1 + \alpha)\rho.$$

To get ρ we try all values

$$\rho := (1+\alpha)^k$$

for $k \in \mathbb{Z}$ satisfying $k_0 \leq k \leq k_1$, where

$$\begin{aligned} k_0 &:= \log_{1+\alpha}(r^{-(n^2+n)}2^{-c(n)}) \text{ and} \\ k_1 &:= \log_{1+\alpha}\left(n \cdot \max\left\{||b_i||\ |1 \leq i \leq m\right\}\right), \end{aligned}$$

where $[b_1, \ldots, b_m] \in \mathbb{Q}^{n \times m}$ is a basis of the lattice L and $c \in \mathbb{Z}[X]$ is a polynomial satisfying $2^{-c(n)}\|x\|_2 \leq \|x\| \leq 2^{c(n)}\|x\|_2$ for all $x \in \mathbb{R}^n$. We need to argue that there exists an integer k with $k_0 \leq k \leq k_1$ satisfying $\rho \leq \mu^{(\|\cdot\|)}(t, L) < (1+\alpha)\rho$.

Claim 4.3.5. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$ and $c \in \mathbb{Z}[X]$ be a polynomial such that $2^{-c(n)}\|x\|_2 \leq \|x\| \leq 2^{c(n)}\|x\|_2$ for all $x \in \mathbb{R}^n$. Let $L \subseteq \mathbb{Q}^n$ be a lattice and $t \in \operatorname{span}(L) \cap \mathbb{Q}^n$ be some target vector satisfying $t \notin L$. Then*

$$\mu^{(\|\cdot\|)}(t, L) \geq r^{-(n^2+n)}2^{-c(n)},$$

where r is an upper bound on the size of the basis defining the lattice and the target vector.

Proof. We can transform the lattice $L \subseteq \mathbb{Q}^n$ and the target vector $t \in \mathbb{Q}^n$ into a lattice $\tilde{L} \subseteq \mathbb{Z}^n$ and a target vector $\tilde{t} \in \mathbb{Z}^n$ by doing the following: We multiply the basis and the target vector t with the least common multiple lcm of the at most n^2 denominators of the coefficients of the basis vectors and the n denominators of the coefficients of the target vector t. This means, we multiply each coefficient with an integer of size at most r^{n^2+n}, since r is an upper bound on the size of the basis vectors and the target vector. Obviously, $\tilde{t} \notin \tilde{L}$. Since $\tilde{L} \subseteq \mathbb{Z}^n$ and $\tilde{t} \in \mathbb{Z}^n$, the Euclidean distance between the target vector $\tilde{t}$ and the lattice $\tilde{L}$ is at least 1, $\mu^{(2)}(\tilde{t}, \tilde{L}) \geq 1$. Since $\|\cdot\|$ is a tractable norm, the distance between $\tilde{t}$ and $\tilde{L}$ with respect to the norm $\|\cdot\|$ is at least $\mu^{(\|\cdot\|)}(\tilde{t}, \tilde{L}) \geq 2^{-c(n)}$. This implies that

$$\mu^{(\|\cdot\|)}(t, L) \geq r^{-(n^2+n)}2^{-c(n)}.$$

□

Using a standard rounding argument, we can see that for every target vector $t \in \operatorname{span}(L)$ its distance to the lattice is at most $m \cdot \max\{\|b_i\||1 \leq i \leq m\}$, where m is the rank of the lattice and $B = [b_1, \ldots, b_m]$ is a basis of L:

Claim 4.3.6. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{R}^n$ be a lattice given by a basis $B = [b_1, \ldots, b_m]$ and $t \in \operatorname{span}(L)$. Then the distance between the target vector t and the lattice L is at most*

$$\mu^{(\|\cdot\|)}(t, L) \leq \frac{1}{2}\sum_{i=1}^{m}\|b_i\| \leq m \cdot \max\left\{\|b_j\| \mid 1 \leq j \leq m\right\}.$$

Proof. Since $t \in \operatorname{span}(L)$, there exists a representation of t as a linear combination of the basis vectors, $t = \sum_{i=1}^{n} t_i b_i$ with $t_i \in \mathbb{R}$ for all $1 \leq i \leq m$. The distance between t and the lattice vector

$$\sum_{i=1}^{m} \lfloor t_i \rceil b_i \in \mathcal{L}(B) = L,$$

which is given by rounding each coefficient of t to the nearest integer, is bounded by

$$\left\| t - \sum_{i=1}^{m} \lfloor t_i \rceil b_i \right\| \leq \sum_{i=1}^{m} |t_i - \lfloor t_i \rceil| \cdot \|b_i\| \leq \frac{1}{2} \sum_{i=1}^{m} \|b_i\|.$$

□

Combining Claim 4.3.5 and Claim 4.3.6 we obtain

$$r^{-(n^2+n)} 2^{-c(n)} \leq \mu^{(\|\cdot\|)}(t, L) \leq n \cdot \max\{||b_i|| \mid 1 \leq i \leq m\},$$

where r is an upper bound on the size of the basis B defining the lattice L and the target vector and $c \in \mathbb{Z}[X]$ is a polynomial satisfying $2^{-c(n)}\|x\|_2 \leq \|x\| \leq 2^{c(n)}\|x\|_2$ for all $x \in \mathbb{R}^n$. Therefore, there exists an integer k with $k_0 \leq k \leq k_1$ satisfying $(1+\alpha)^k \leq \mu^{(\|\cdot\|)}(t, L) < (1+\alpha)^{k+1}$. Moreover, we only need to try polynomially (in $\log_2(r)$ and $1/\alpha$) many guesses of the form $r := (1+\alpha)^k$.

In the following, we assume that the parameter ρ satisfies $\rho \leq \mu^{(\|\cdot\|)}(t, L) < (1+\alpha)\rho$. For $0 \leq \epsilon \leq 1/2$ we define the parameter γ as

$$\gamma := \frac{1+\epsilon}{1-\epsilon}(1+\alpha)\rho. \tag{4.4}$$

We consider the lattice $L' \subseteq \mathbb{R}^{n+1}$ and the subspace $M \in \mathbb{R}^{n+1}$, defined as

$$L' := \mathcal{L}\left(\left[\begin{pmatrix} b_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} b_m \\ 0 \end{pmatrix}, \begin{pmatrix} t \\ \gamma \end{pmatrix}\right]\right) \text{ and} \tag{4.5}$$

$$M := \operatorname{span}\left(\left\{\begin{pmatrix} b_i \\ 0 \end{pmatrix} |1 \leq i \leq m\right\}\right) \subsetneq \operatorname{span}(L'), \tag{4.6}$$

where $[b_1, \ldots, b_m] \in \mathbb{Q}^{n \times m}$ is a basis of the lattice L.

First of all we give an upper bound on the subspace avoiding minimum of this Gsvp-instance.

Claim 4.3.7. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$ and F be defined as in (4.3). Let $0 \leq \epsilon \leq 1/2$, let $L \subseteq \mathbb{R}^n$ be a lattice and let $t \in \operatorname{span}(L)$ be some target vector. For $\alpha > 0$ arbitrary let ρ be a parameter satisfying $\rho < \mu^{(\|\cdot\|)}(t, L) \leq (1+\alpha)\rho$. Let the parameter γ, the lattice $L' \subseteq \mathbb{R}^{n+1}$ and the subspace $M \subsetneq \operatorname{span}(L')$ be defined as above, see (4.4), (4.5), and (4.6). Then the subspace avoiding minimum of L and M with respect to the norm F is less than*

$$\lambda_M^{(F)}(L') < \frac{2}{1+\epsilon}\gamma.$$

Proof. Let $z \in L$ be the lattice vector that is closest to the target vector t with respect to the norm $\|\cdot\|$. Then $(z-t,-\gamma) \in L' \backslash M$ and the length of the vector $(z-t,-\gamma)$ with respect to the norm F is bounded by

$$\begin{aligned} F\left((z-t,-\gamma)\right) &= ||z-t|| + |\gamma| \\ &= \mu^{(\|\cdot\|)}(t,L) + |\gamma| \\ &< (1+\alpha)\rho + \frac{1+\epsilon}{1-\epsilon}(1+\alpha)\rho \\ &= \frac{2}{1-\epsilon}(1+\alpha)\rho \\ &= \frac{2}{1+\epsilon}\gamma. \end{aligned}$$

Therefore, the subspace avoiding minimum is smaller than $\lambda_M^{(F)}(L') < 2\gamma/(1+\epsilon)$. □

The following lemma shows that given an oracle $\mathcal{A}$ that solves the generalized shortest vector problem with respect to the norm F defined in (4.3) exactly, we can solve the closest vector problem with respect to the norm $\|\cdot\|$ exactly.

Lemma 4.3.8. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$ and F be defined as in (4.3). Let $0 \leq \epsilon \leq 1/2$, let $L \subseteq \mathbb{R}^n$ be a lattice and let $t \in \operatorname{span}(L)$ be some target vector. For $\alpha > 0$ arbitrary let ρ be a parameter satisfying $\rho < \mu^{(\|\cdot\|)}(t,L) \leq (1+\alpha)\rho$. Let the parameter γ, the lattice $L' \subseteq \mathbb{R}^{n+1}$ and the subspace $M \subsetneq \operatorname{span}(L')$ be defined as above, see (4.4), (4.5), and (4.6).*
If $u \in L' \backslash M$ with $F(u) = \lambda_M^{(F)}(L')$, then $u = \pm(z-t,-\gamma)$ where $z \in L$ is a lattice vector that is closest to the target vector t with respect to the norm $\|\cdot\|$.

Proof. We have seen in Claim 4.3.7 that with our assumptions the subspace avoiding minimum of the lattice L' and the subspace M with respect to the norm F is less than 2γ,

$$\lambda_M^{(F)}(L') < 2\gamma.$$

Hence, the vector u is of the form $u = (z \pm t, \pm\gamma)$ for some lattice vector $z \in L$. Therefore, $|| \mp z - t|| = \mu^{(\|\cdot\|)}(t,L)$ and $\mp z$ is a lattice vector closest to t. □

This result shows that there exists a polynomial time reduction from the exact version of the closest vector problem to the exact version of the generalized shortest vector problem.

Next, we assume that the oracle $\mathcal{A}$ solves the generalized shortest vector problem with approximation factor $0 < \epsilon \leq 1/2$. Because of the lifting technique we are not able to solve the closest vector problem with an approximation factor $1+\epsilon$ but only with an approximation factor $(1+4\epsilon)(1+\alpha)$.

Lemma 4.3.9. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$ and F be defined as in (4.3). Let $0 < \epsilon \leq 1/2$, let $L \subseteq \mathbb{R}^n$ be a lattice and let $t \in \operatorname{span}(L)$ be some target vector. For $\alpha > 0$ arbitrary let ρ be a parameter satisfying $\rho < \mu^{(\|\cdot\|)}(t, L) \leq (1+\alpha)\rho$. Let the parameter γ, the lattice $L' \subseteq \mathbb{R}^{n+1}$ and the subspace $M \subsetneq \operatorname{span}(L')$ be defined as above, see (4.4), (4.5), and (4.6).*
Let $v \in L' \backslash M$ be a vector satisfying $F(v) < (1+\epsilon)\lambda_M^{(F)}(L')$. Then a lattice vector $z^* \in L$ with

$$||z^* - t|| \leq (1+4\epsilon)(1+\alpha)\mu^{(\|\cdot\|)}(t, L)$$

can be computed in polynomial time.

Proof. Since the subspace avoiding minimum $\lambda_M^{(F)}(L')$ is less than $(2/(1+\epsilon))\gamma$, see Claim 4.3.7, the length of the vector v with respect to the norm F is at most

$$F(v) < (1+\epsilon)\lambda_M^{(F)}(L') < 2\gamma. \tag{4.7}$$

Since $v \in L' \backslash M$, the vector v is of the form $v = \pm(z^* - t, -\gamma)$ for some lattice vector $z^* \in L$. Without loss of generality we assume $v = (z^* - t, -\gamma)$. Hence,

$$F(v) = ||z^* - t|| + \gamma \tag{4.8}$$

and we can give an upper bound on the distance between the lattice vector z^* and the target vector t with respect to the norm $\|\cdot\|$,

$$\|z^* - t\| = F(v) - \gamma < 2\gamma - \gamma = \gamma,$$

using Inequality (4.7). The parameter γ is defined as $\gamma = ((1+\epsilon)/(1-\epsilon))(1+\alpha)\rho$, see Equation (4.4). Since we assume that $\rho < \mu^{(\|\cdot\|)}(t, L)$ this is less than

$$\gamma < \frac{1+\epsilon}{1-\epsilon}(1+\alpha)\mu^{(\|\cdot\|)}(t, L).$$

Using the inequality $1/(1-\epsilon) \leq 1 + 2\epsilon$ which holds for all $\epsilon < 1/2$, we obtain

$$\gamma < (1+2\epsilon)(1+\epsilon)(1+\alpha)\mu^{(\|\cdot\|)}(t, L) \leq (1+4\epsilon)(1+\alpha)\mu^{(\|\cdot\|)}(t, L).$$

This shows that the vector $z^* \in L$ is an $(1+4\epsilon)(1+\alpha)$-approximation of the closest lattice vector to t with respect to the norm $\|\cdot\|$. □

Summarizing, we get

Theorem 4.3.10. *For all tractable norms, the exact version of* Cvp *is polynomial time reducible to the exact version of* Gsvp. *Also, for all efficiently computable norms,* Cvp *with approximation factor $(1+\epsilon)(1+\alpha)$ for $0 < \epsilon \leq 1/2$ and $\alpha > 0$ is reducible to* Gsvp *with approximation factor $1+\epsilon/4$. The reduction is polynomial in the representation size of the* Cvp *instance and in $1/\alpha$.*

If we want to solve the closest vector problem with respect to an ℓ_p-norm for $1 \leq p \leq \infty$ using an oracle for the generalized shortest vector problem, the reduction described above can be simplified using that for fixed p, the ℓ_p-norms are a family of norms. That means for all $n \in \mathbb{N}$, the function $\mathbb{R}^n \to \mathbb{R}^{\geq 0}$, $x \mapsto (\sum_{i=1}^n |x_i|^p)^{1/p}$ defines a norm on $\mathbb{R}^n$ called the ℓ_p-norm on $\mathbb{R}^n$. Hence, we do not need to solve the generalized shortest vector problem with respect to the norm F as defined in (4.3). Instead we solve the generalized shortest vector problem with respect to the ℓ_p-norm on $\mathbb{R}^{n+1}$. In this case, we can use the same construction for the reduction from the closest vector problem to the generalized shortest vector problem as above but with the parameter

$$\gamma := \frac{1}{\sqrt[p]{2^p - (1+\epsilon)^p}}(1+\epsilon)(1+\alpha)\rho$$

for $1 \leq \rho < \infty$. For the ℓ_∞-norm, we set

$$\gamma := \frac{1}{2}(1+\epsilon)(1+\alpha)\rho.$$

Then we obtain

Theorem 4.3.11. *For all ℓ_p-norms with $1 \leq p \leq \infty$, the exact version of the closest vector problem in the ℓ_p-norm is polynomial time reducible to the exact version of the generalized shortest vector problem in the ℓ_p-norm.*
Also, for all ℓ_p-norms with $1 \leq p \leq \infty$, the closest vector problem in the ℓ_p-norm with approximation factor $(1+\epsilon)(1+\alpha)$ for $0 < \epsilon \leq 1/2$ and $\alpha > 0$ is reducible to the generalized shortest vector problem in the ℓ_p-norm with approximation factor $1 + \epsilon/6$. The reduction is polynomial time in the representation size of the Cvp*-instance and in $1/\alpha$.*

The proof of this theorem can be found in [BN09].

4.3.2. The lattice membership problem

In this section we give a geometric reformulation of the closest vector problem. We will use this different point of view on the closest vector problem to present in Chapter 6 a deterministic polynomial space bounded algorithm for this lattice problem. Since there exist polynomial time reductions from Svp, Smp, and Sivp to Cvp, we also obtain deterministic polynomially space bounded algorithms for the other lattice problems, see Figure 4.10.

For the reformulation, we use the equivalence between norms and convex bodies which we considered already in Chapter 2 of this thesis, see Section 2.1. We reformulate the closest vector problem as a membership problem for certain convex sets.

Definition 4.3.12. *(Lattice membership problem (*Lmp*))*
Given a lattice $L \subseteq \mathbb{R}^n$ and a bounded convex set $\mathcal{C} \subseteq \operatorname{span}(L)$, output a lattice vector in $\mathcal{C}$ or decide that $\mathcal{C}$ does not contain a vector from L.

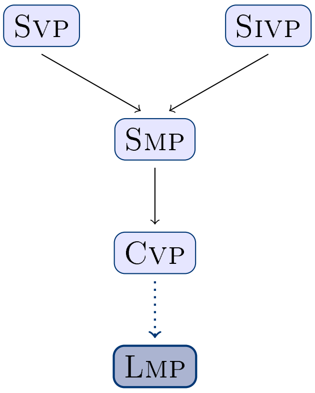

Figure 4.10.: **Relation between LMP and the other lattice problems.** Arrows indicate polynomial time reductions preserving the rank of the lattice and the approximation factor. The arrow from CVP to LMP is marked dotted since this reduction works only for the exact version of CVP.

The lattice membership problem is a generalization of the integer programming feasibility problem from polyhedra to bounded convex sets. In the integer programming feasibility problem we are given a polyhedron and the goal is to decide whether this polyhedron contains an integer vector. It is known that the integer programming feasibility problem in NP-complete, see [Coo71].

There is a strong relation between the lattice membership problem and the decisional variant of the closest vector problem. As already mentioned, in the decisional closest vector problem we are given a lattice $L \subseteq \mathbb{R}^n$, some target vector $t \in \operatorname{span}(L)$ and a parameter $\alpha > 0$. The goal is to decide whether the distance between the target vector and the lattice is at most α or not. Obviously, the decisional closest vector problem can be seen as a special case of the lattice membership problem where the corresponding convex set is the ball $\bar{B}_n^{(\|\cdot\|)}(t, \alpha)$ and where we obtain an additional certificate if the distance between the target vector t and the lattice is at most α.

In this section we show that if we are able to solve the lattice membership problem for balls generated by a norm, we are able to solve the closest vector problem with respect to this norm. As already mentioned, we can assume in the following that we consider a lattice $L \subseteq \mathbb{Z}^n$ and some target vector $t \in \operatorname{span}(L) \cap \mathbb{Z}^n$.

Theorem 4.3.13. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Assume that there exists an algorithm $\mathcal{A}$ that for all lattices $\mathcal{L}(B') \subseteq \mathbb{Z}^n$ of rank m and all target vectors $t' \in \operatorname{span}(B) \cap \mathbb{Z}^n$ solves the lattice membership problem for the ball $B_n^{(\|\cdot\|)}(t', \alpha)$ using at most $T_{m,n}^{(\|\cdot\|)}(r', \alpha)$ arithmetic operations. Here r' is an upper bound on the size of the basis B' and the target vector t'.*

- *If the norm is an ℓ_p-norm with $1 \leq p \leq \infty$, there exists an algorithm $\mathcal{A}'$ that* *solves the closest vector problem for all lattices $\mathcal{L}(B) \subseteq \mathbb{Z}^n$ and target vectors $t \in \operatorname{span}(B) \cap \mathbb{Z}^n$. The number of arithmetic operations of this algorithm is*

$$k \cdot n^{\mathcal{O}(1)} \log_2(r)^2 \cdot T^{(p)}_{m,n}(r, mn^{3/2} r),$$

where $k = p$ for $1 \leq p < \infty$ and $k = 1$ for $p = \infty$.

- *If the norm is a polyhedral norm given by a full-dimensional polytope symmetric about the origin with s constraints, then there exists an algorithm that solves the closest vector problem for all lattice $\mathcal{L}(B) \subseteq \mathbb{Z}^n$ and target vectors $t \in \operatorname{span}(B) \cap \mathbb{Z}^n$. The number of arithmetic operations of this algorithm is*

$$s \cdot n^{\mathcal{O}(1)} \log_2(\operatorname{size}(P) \cdot r) \cdot T^{(P)}_{m,n}(r, nmr \operatorname{size}(P)).$$

In both cases, r is an upper bound on the size of the basis B and the target vector t.

The proof of this result is a variant of the proof that all three variants of the closest vector problem are equivalent. For the closest vector problem with respect to the Euclidean norm this was shown by Micciancio and Goldwasser, see [MG02] and [Mic07]. Their result can be generalized to arbitrary ℓ_p-norms, $1 \leq p \leq \infty$, and to polyhedral norms, see [BN11].

The reduction from the closest vector problem to the lattice membership problem is based on binary search. This binary search is performed on the set of all possible values which can be achieved by the norm of an integer vector if the norm lies in some certain interval. Hence, we need to ensure that we are able to enumerate all these values and we need an upper bound on the cardinality of such a set - depending on the size of the interval. To guarantee all that, we consider special norms which we call enumerable. In general, we call a function enumerable if it maps every integer vector to a discrete enumerable set.

Definition 4.3.14. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called (k, K)-enumerable for parameters $k, K \in \mathbb{N}$ or simply enumerable if there exists $\tilde{K} \in \mathbb{N}$ with $\tilde{K} \leq K$, such that*

$$\tilde{K} \cdot f(x)^k \in \mathbb{N}_0 \text{ for all } x \in \mathbb{Z}^n.$$

Obviously, every ℓ_p-norm,$1 \leq p \leq \infty$, is $(k, 1)$-enumerable with $k = p$ for $1 \leq p < \infty$ and $k = 1$ for $p = \infty$. Later, we will show that also all polyhedral norms are enumerable. In contrast, the function over $\mathbb{R}$ which maps every number to its inverse in absolute values,

$$\mathbb{R} \to \mathbb{R},\ x \mapsto \begin{cases} 1/|x| & , x \neq 0 \\ 0 & , x = 0 \end{cases}$$

is not enumerable. For all $K \in \mathbb{N}$ there exists an integer whose image is not contained in $(1/K) \cdot \mathbb{N}_0$, for example $1/(K+1) \notin (1/K) \cdot \mathbb{N}_0$.

For all (k, K)-enumerable norms which are efficiently computable we are able to give a reduction from the closest vector problem to the lattice membership problem. The number of arithmetic operations depends on the parameters k and K.

Proposition 4.3.15. *Let $\|\cdot\|$ be a (k, K)-enumerable norm on $\mathbb{R}^n$ which is efficiently computable. Assume that there exists an algorithm $\mathcal{A}$ that for all lattices $\mathcal{L}(B') \subseteq \mathbb{Z}^n$ of rank m, all balls $B_n^{(\|\cdot\|)}(t', \alpha)$ with $t' \in \operatorname{span}(B') \cap \mathbb{Z}^n$ and $\alpha > 0$, solves the lattice membership problem. Let $T_{m,n}^{(\|\cdot\|)}(r', \alpha)$ be an upper bound on the number of arithmetic operations of $\mathcal{A}$ where r' is an upper bound on the size of the lattice basis B' and the vector t'.*
Then, there exists an algorithm that solves the closest vector problem for all lattices $\mathcal{L}(B) \subseteq \mathbb{Z}^n$ and all target vectors $t \in \operatorname{span}(B) \cap \mathbb{Z}^n$ with respect to the norm $\|\cdot\|$. The number of arithmetic operations of this algorithm is

$$\big(k \cdot \log_2(m \cdot \|b\|) + \log_2(K)\big) \cdot n^{\mathcal{O}(1)} \cdot T_{m,n}^{(\|\cdot\|)}(r, m \cdot \|b\|),$$

where r is an upper bound on the size of the Cvp*-instance $(\mathcal{L}(B), t)$ and $\|b\|$ is an upper bound on the length of each basis vector of the basis B, $\|b\| := \max\{\|b_j\| | 1 \leq j \leq m\}$. Each number computed by the algorithm has size of at most*

$$\max\{m \cdot \|b\|, K\}^k$$

Proof. Let $B = [b_1, \ldots, b_m] \subseteq \mathbb{Z}^{n \times m}$ be a lattice basis of the lattice L and $t \in \operatorname{span}(L) \cap \mathbb{Z}^n$ be some target vector. Without loss of generality, we assume that $t \notin L$, i.e., $\mu^{(\|\cdot\|)}(t, L) > 0$. Since $t \in \operatorname{span}(L)$, we can choose

$$R := m \cdot \max\{\|b_j\| | 1 \leq j \leq m\}$$

as an upper bound for the distance between the target vector and the lattice, see Claim 4.3.6.

We have $L \subseteq \mathbb{Z}^n$ and $t \in \mathbb{Z}^n$. Hence, the distance vector of t and its closest lattice vector is an integer vector. Using that $\|\cdot\|$ is a (k, K)-enumerable norm, we obtain that the distance between t and the lattice is of the form

$$\mu^{(\|\cdot\|)}(t, L) = \sqrt[k]{\frac{p}{q}}, \text{ where } p, q \in \mathbb{N} \text{ with } \gcd(p, q) = 1 \text{ and } 1 \leq q \leq K.$$

Since R is an upper bound on the distance between the vector t and the lattice, we have $p/q \leq R^k$.

Now, we perform a binary search on the interval $[0, R^k]$. We start by calling the algorithm $\mathcal{A}$ with input of the lattice $\mathcal{L}(B)$ and the convex set $B_n^{(\|\cdot\|)}(t, R/\sqrt[k]{2})$. Either the algorithm computes a lattice vector in this ball or it decides that $B_n^{(\|\cdot\|)}(t, R/\sqrt[k]{2})$ does not contain a lattice vector. Depending on the answer, we continue in the usual way.

Suppose we have found two radii $r_1 > r_0 > 0$ such that $B_n^{(\|\cdot\|)}(t, r_0)$ does not contain a lattice vector, whereas the convex set $B_n^{(\|\cdot\|)}(t, r_1)$ contains a lattice vector $v \in L$. If the difference between r_0 and r_1 is less than $1/K^2$, then $v \in L$ is a closest lattice vector to t: In an interval of length less than $1/K^2$ there exists at most one number of the form p/q with $\gcd(p, q) = 1$ and $1 \leq q \leq K$.
Since $v \in L \subseteq \mathbb{Z}^n$, the norm of v is the k-th root of such a number, $\|v - t\|^k = p/q$ with $p, q \in \mathbb{N}$, $\gcd(p, q) = 1$ and $1 \leq q \leq K$. Hence, $v \in L$ is a lattice vector with $\|v - t\| = \mu^{(\|\cdot\|)}(t, L)$.

The number of calls to the algorithm $\mathcal{A}$ is at most $\mathcal{O}(\log_2(R^k \cdot K^2))$, since we are finished if the length of the current interval is less than $1/K^2$. As a consequence, the number of arithmetic operations needed to solve the closest vector problem is

$$\mathcal{O}\big(k \cdot \log_2(R) + 2\log_2(K)\big) \cdot n^{\mathcal{O}(1)} \cdot T_{m,n}^{(\|\cdot\|)}(S, R).$$

Since the distance between the target vector and the lattice is of the form $\sqrt[k]{p}/\sqrt[k]{q}$, where $p, q \in \mathbb{N}$ with $1 \leq q \leq K$ and $p \leq m \cdot \max\{\|b_j\| | 1 \leq j \leq m\} = m \cdot \|b\|$, each number computed by the algorithm has size at most $\max\{m \cdot \|b\|, K\}^k$. □

Now we want to apply this result to ℓ_p-norms, $1 \leq p \leq \infty$, and polyhedral norms. The corresponding result for all ℓ_p-norms follows directly from a special case of Hölder's inequality, see Proposition 2.2.15 in Chapter 2.

Corollary 4.3.16. *For all ℓ_p-norms with $1 \leq p \leq \infty$, assume that there exists an algorithm $\mathcal{A}$ that solves the lattice membership problem for all lattices $\mathcal{L}(B') \subseteq \mathbb{Z}^n$ of rank m and balls $B_n^{(\|\cdot\|)}(t', \alpha)$, where $t' \in \operatorname{span}(B') \cap \mathbb{Z}^n$ and $\alpha > 0$. The number of arithmetic operations of the algorithm is at most $T_{m,n}^{(\|\cdot\|)}(r', \alpha)$, where r' is an upper bound on the size of the basis B' and the target vector t'.*
Then, there exists an algorithm $\mathcal{A}'$, that solves the closest vector problem for all lattices $\mathcal{L}(B) \subseteq \mathbb{Z}^n$. The number of arithmetic operations of the algorithm $\mathcal{A}'$ is at most

$$k \cdot n^{\mathcal{O}(1)} \log_2(r) T(r, mn^{3/2} r),$$

where $k = p$ for $1 \leq p < \infty$ and $k = 1$ for $p = \infty$. Here, r is an upper bound on the size of the basis B and the target vector t.

Proof. Obviously, all ℓ_p-norms are $(k, 1)$ enumerable with $k = p$ for $1 \leq p < \infty$ and $k = 1$ for $p = \infty$. Hence, it follows from Proposition 4.3.15 that there exists an algorithm that solves the closest vector problem in any ℓ_p-norm. The number of arithmetic operations of this algorithm is at most

$$\log_2(r) k \cdot \log_2(\|b\|_p) n^{\mathcal{O}(1)} \cdot T_{m,n}^{(p)}(r, m \cdot \|b\|),$$

where $\|b\|_p$ is an upper bound on the length of the basis vectors. The length of the basis vectors is upper bounded by

$$\|b_i\|_p \leq n \cdot \|b_i\|_2 \leq n^{3/2} \operatorname{size}(B) = n^{3/2} \cdot r,$$

see Claim 2.2.18 in Chapter 2. This shows that the statement is correct. □

To get the corresponding result for polyhedral norms, we need to show that all polyhedral norms are enumerable. This is done in the following lemma.

Lemma 4.3.17. *Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope symmetric about the origin given by s constraints, i.e., $P = \{x \in \mathbb{R}^n | \langle x, h_i \rangle \leq \beta_i$ and $\langle x, -h_i \rangle \leq \beta_i$ for all $1 \leq i \leq s/2\}$, where $h_1, \dots, h_{s/2} \in \mathbb{Z}^n$ and $\beta_1, \dots, \beta_{s/2} \in \mathbb{N}$. Then $\| \cdot \|_P$ is a $(1, \prod_{j=1}^{s/2} \beta_j)$-enumerable norm.*

Proof. Given an integer vector $x \in \mathbb{Z}^n \backslash \{0\}$ its polyhedral norm has value r if the following two properties are satisfied:

- The vector x is contained in the scaled polytope $r \cdot P$, that means $\langle x, h_i \rangle \leq r \cdot \beta_i$ and $\langle x, -h_i \rangle \leq \beta_i$ for all $1 \leq i \leq s/2$.
- There exists at least one inequality defining the polytope which is fulfilled with equality. Let $j \in \mathbb{N}$, $1 \leq j \leq s/2$, be such an index. Without loss of generality, we assume that $\langle x, h_j \rangle = r \cdot \beta_j$. Since $\langle x, h_j \rangle \in \mathbb{Z}$, we have $r = \langle x, h_j \rangle / \beta_j \in \mathbb{Q}$. That means, there exists $p, q \in \mathbb{N}$ with $\gcd(p, q) = 1$ such that $r = p/q$. Additionally, we know that β_j is divisible by q.

That means, that each value, which can be achieved by the norm $\| \cdot \|_P$ of an integer vector, is a rational of the form p/q with $p, q \in \mathbb{N}$ and $\gcd(p, q) = 1$, and there exists an index j, $1 \leq j \leq s/2$, such that q divides β_j. Hence, for each vector $x \in \mathbb{Z}^n$ we obtain that $(\prod_{j=1}^{s/2} \beta_j) \cdot \|x\|_P \in \mathbb{N}_0$. □

Corollary 4.3.18. *Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope symmetric about the origin given by s constraints. Assume that there exists an algorithm $\mathcal{A}$ that solves the lattice membership problem for all lattices $\mathcal{L}(B') \subseteq \mathbb{Z}^n$ of rank m and all convex sets $B_n^{(P)}(t', \alpha)$, where $t' \in \operatorname{span}(B') \cap \mathbb{Z}^n$ and $\alpha > 0$ using at most $T_{m,n}^{(P)}(r, \alpha)$ arithmetic operations, where r' is an upper bound on the size of the basis B' and the vector t'.*
Then there exists an algorithm $\mathcal{A}'$ that solves the closest vector problem with respect to the polyhedral norm $\| \cdot \|_P$ for all lattices $\mathcal{L}(B) \subseteq \mathbb{Z}^n$ of rank m and target vectors $t \in \operatorname{span}(B) \cap \mathbb{Z}^n$ in time

$$s \cdot n^{\mathcal{O}(1)} \log_2(\operatorname{size}(P) \cdot r) \cdot T_{m,n}^{(P)}(r, n \cdot m \cdot r \cdot \operatorname{size}(P)),$$

where r is an upper bound on the size of the basis B and the target vector t.

Proof. Assume that P is given by a set $H_P = \{h_1, \dots, h_{s/2}\} \subseteq \mathbb{Z}^n$ and a set of parameters $\{\beta_1, \dots, \beta_{s/2}\} \subseteq \mathbb{N}$, i.e., $P = \{x \in \mathbb{R}^n | \langle x, h_i \rangle \leq \beta_i$ and $\langle x, -h_i \rangle \leq \beta_i$ for all $1 \leq i \leq s/2\}$.
We have seen in Lemma 4.3.17 that the norm $\| \cdot \|_P$ defined by the polytope P is $(1, \prod_{j=1}^{s/2} \beta_j)$-enumerable. Since the parameters β_j, $1 \leq j \leq s/2$, are integers, we have

$$\prod_{j=1}^{s/2} \beta_j \leq \operatorname{size}(P)^{s/2}.$$

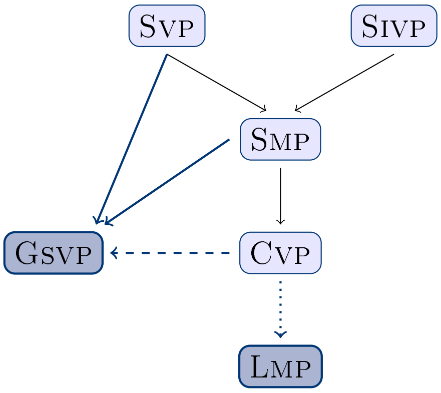

Figure 4.11.: **Relations among the lattice problems that will be used in this thesis.** Arrows indicate polynomial time reductions preserving the rank of the lattice and the approximation factor. The arrow from CVP to GSVP is marked dashed since the approximation factor is not exactly preserved by the reduction. The arrow from CVP to LMP is marked dotted since this reductions works only for the exact version of CVP.

Hence, it follows from Proposition 4.3.15 that there exists an algorithm $\mathcal{A}'$ that solves the closest vector problem with respect to the polyhedral norm $\|\cdot\|_P$ and the number of arithmetic operations of this algorithm is at most

$$\big(\log_2(\|b\|_P) + r \cdot \log_2(\text{size}(P))\big) n^{\mathcal{O}(1)} \cdot T(r, m \cdot \|b\|_P),$$

where $\|b\|_P$ is an upper bound on the length of the basis vectors. As we have seen in Corollary 2.2.20, we have $\|x\|_P \leq n^{(n+1)/2}\,\text{size}(P)^n \|x\|_2$. Thus, for all $1 \leq i \leq m$ we have

$$\|b_i\|_P \leq n^{(n+1)/2}\,\text{size}(P)^n \|b_i\|_2 \leq n^{(n+1)/2}\,\text{size}(P)^n \cdot \sqrt{n} \cdot r,$$

where r is an upper bound on the size of the basis B defining the lattice, see Claim 2.2.18 in Chapter 2. This shows that the statement is correct. □

Now the proof of Theorem 4.3.13 follows directly from Corollary 4.3.16 and Corollary 4.3.18.

Perspective

At this point, we have two starting points for the development of lattice algorithms, the generalized shortest vector problem and the lattice membership problem, see Figure 4.11. Based on these results we will present in the rest of this thesis essentially two different types of lattice algorithms.

In the next chapter, we describe randomized single exponential time algorithms for the generalized shortest vector problem for all tractable norms. The algorithms are based on a sampling technique developed by Ajtai, Kumar, and Sivakumar in 2001. This technique is called the AKS-sampling technique. The first algorithm described in Chapter 5 approximates the generalized shortest vector problem with approximation factor $1+\epsilon$ for arbitrary $0 < \epsilon < 3/2$. Combining this algorithm with the reductions presented in this chapter, we obtain corresponding single exponential time approximation algorithms for Svp, Smp, Sivp, and Cvp.

By slightly modifying our algorithm for the generalized shortest vector problem we obtain an algorithm that solves the generalized shortest vector problem exactly but only for instances where there do not exist too many short lattice vectors outside the given subspace. As a consequence, we obtain algorithms that solve the four lattice problems Svp, Smp, Sivp, and Cvp exactly but only for instances where there do not exist too many approximate solutions. As we have seen in this chapter, this can only be guaranteed for the shortest vector problem, see Corollary 4.2.12. Thus, for Smp, Sivp, and Cvp we do not obtain algorithms which solves these problems exactly. Another disadvantage of these algorithms based on the AKS-sampling technique is that they need exponential space.

For this reason we present in Chapter 6 a deterministic polynomially space bounded algorithm for the lattice membership problem for polytopes and ℓ_p-balls. Compared to our algorithms for the generalized shortest vector problem, the number of arithmetic operations of our algorithms is not single exponential in the dimension n but mainly determined by the factor $n^{(2+o(1))n}$.

As we have seen, there exists a polynomial time reduction from the closest vector problem to the lattice membership problem which works for all tractable norms. Hence, we obtain a deterministic polynomially space bounded algorithm for Cvp which works for all ℓ_p-norms, $1 \leq p < \infty$, and all polyhedral norms, in particular for the ℓ_1-norm and the ℓ_∞-norm.

Obviously, we obtain also deterministic polynomially space bounded algorithms for the other lattice problems Svp, Smp, and Sivp, since for all these problems there exist polynomial time reductions to the closest vector problem. Of course, for Svp this result is not really interesting.

5. A randomized algorithm for the generalized shortest vector problem

In this chapter, we present a probabilistic single exponential time algorithm for the generalized shortest vector problem for all tractable norms. To recall, in the generalized shortest vector problem we are given a lattice L together with some subspace $M \subsetneq \operatorname{span}(L)$ and we are asked to find a shortest lattice vector in $L \backslash M$, see Definition 4.3.1 in Chapter 4. The algorithm solves the generalized shortest vector problem almost optimally, i.e., with approximation factor $1 + \epsilon$ for arbitrary $0 < \epsilon < 3/2$. Additionally, we present a probabilistic single exponential time algorithm that solves a restricted version of the generalized shortest vector problem optimally. We describe these algorithms only for full-dimensional lattices. However, our results can easily be generalized to arbitrary lattices.

We have already seen in Chapter 4 that there are polynomial time reductions from the shortest vector problem, the closest vector problem, the successive minima problem, and the shortest independent vectors problem to the generalized shortest vector problem. These reductions establish probabilistic single exponential time algorithms for all these four lattice problems. For Svp and restricted versions of Cvp, Smp, and Sivp, we obtain algorithms that solve these problems optimally. For the general versions of Cvp, Smp, and Sivp, we obtain algorithms with approximation factor $1+\epsilon$ for arbitrary $0 < \epsilon < 3/2$.

The AKS-sampling technique

Prior to the breakthrough paper [AKS01] of Ajtai, Kumar and Sivakumar, randomization has rarely been utilized in algorithms for lattice problems.[1] In their paper from 2001, they describe a novel sampling technique that generates short vectors from the input lattice.

The AKS-sampling method for Svp and Cvp In their paper from 2001, Ajtai, Kumar and Sivakumar describe the first probabilistic algorithm that solves the shortest vector problem with respect to the Euclidean norm optimally with probability exponentially close to 1. More precisely, the number of arithmetic operations used by their algorithm is $(2^n \cdot \log_2(r))^{\mathcal{O}(1)}$, where n is the rank of the lattice and r is an upper bound on the size of the lattice. In particular, the number of arithmetic operations of this algorithm is

[1] An exception is the algorithm of Klein presented in [Kle00] that is a heuristic algorithm for the closest vector problem with respect to the Euclidean norm. The disadvantage of this algorithm is that its running time depends on the distance between the target vector and the lattice.

single exponential only in the rank of the lattice. However, the space complexity of their algorithm is single exponential.

The AKS-algorithm from 2001 was improved by Nguyen and Vidick, Micciancio and Voulgaris, and Pujol and Stehlé, see [NV08], [MV10b], [PS09]. The number of arithmetic operations of the currently fastest AKS-algorithm is $2^{(2.465+o(1))n}\log_2(r)^{\mathcal{O}(1)}$, whereas its space complexity is $2^{(1.233+o(1))n}$.

In 2002, Ajtai, Kumar and Sivakumar extended their sampling technique to solve the closest vector problem with respect to the Euclidean norm with approximation factor $1+\epsilon$ for any $\epsilon > 0$, see [AKS02]. The number of arithmetic operations used by their algorithm is $(2^{(1+1/\epsilon)n}\log_2(r))^{\mathcal{O}(1)}$ and the algorithm is successful with probability exponentially close to 1.

Main results In this chapter, we extend and generalize the results by Ajtai, Kumar and Sivakumar. We show that a variant of the AKS-sampling technique can be used to solve the generalized shortest vector problem. This variant of the AKS-sampling technique is based on a proposal by Sudan and is described in lecture notes by Regev, see [AKS01] and [Reg04]. We obtain an approximation algorithm for the generalized shortest vector problem that works for all tractable norms, i.e., for all efficiently computable norms, for which there exists a polynomial $c \in \mathbb{Z}[X]$ such that $2^{-c(n)}\|x\|_2 \leq \|x\| \leq 2^{c(n)}\|x\|_2$ for all $x \in \mathbb{R}^n$, see Definition 2.1.15 in Chapter 2.

Theorem 5.0.1. *For all tractable norms, there exists a randomized algorithm that approximates the generalized shortest vector problem with success probability* $1 - 2^{-\Omega(n)}$. *The approximation factor is* $1+\epsilon$ *for any* $0 < \epsilon < 3/2$ *and the number of arithmetic operations of the algorithm is* $((2+1/\epsilon)^n \cdot \log_2(r))^{\mathcal{O}(1)}$, *where* n *is the rank of the lattice and* r *is an upper bound on the size of the lattice and the subspace.*

In Chapter 4 we have already seen that there are polynomial time reductions from the shortest vector problem, the closest vector problem, the successive minima problem and the shortest independent vectors problem to the generalized shortest vector problem. Together with Theorem 5.0.1 we obtain a unified treatment for all four lattice problems and single exponential time $(1+\epsilon)$-approximation algorithms for Svp, Cvp, Smp, and Sivp for all tractable norms.

Corollary 5.0.2. *For all tractable norms, there exist randomized algorithms that approximate* Svp, Smp, Sivp, *and* Cvp *with success probability* $1-2^{-\Omega(n)}$. *The approximation factor is* $1+\epsilon$ *for any* $0 < \epsilon < 3/2$ *and the number of arithmetic operations of the algorithm is* $((2+1/\epsilon)^n \cdot \log_2(r))^{\mathcal{O}(1)}$, *where* n *is an upper bound on the rank of the lattice and* r *is an upper bound on the size of the corresponding input instance, i.e., the lattice and perhaps the target vector.*

Next, by slightly modifying the sampling procedure and its analysis, we are able to compute a shortest lattice vector outside a given subspace, provided there do not exist too many short lattice vectors outside the given subspace.

Theorem 5.0.3. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{Q}^n$ be a full-dimensional lattice and $M \subsetneq \operatorname{span}(L)$ be a subspace. Assume that there exist absolute constants c, ϵ such that the number of $v \in L \backslash M$ satisfying $||v|| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ is bounded by 2^{cn}. Then, there exists an algorithm that solves the generalized shortest vector problem with success probability $1 - 2^{-\Omega(n)}$. The number of arithmetic operations of the algorithm is $(2^n \cdot \log_2(r))^{\mathcal{O}(1)}$, where r is an upper bound on the size of the lattice and the subspace. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most $(n \cdot \log_2(r))^{\mathcal{O}(1)}$.*

For the shortest vector problem this requirement is satisfied and we obtain a single exponential time algorithm solving the shortest vector problem optimally.

Theorem 5.0.4. *For all tractable norms, there exists a randomized algorithm that solves the shortest vector problem with success probability $1 - 2^{-\Omega(n)}$. The number of arithmetic operations of the algorithm is $(2^n \log_2(r))^{\mathcal{O}(1)}$, where n is the rank of the lattice and r is an upper bound on its size. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most $(n \cdot \log_2(r))^{\mathcal{O}(1)}$.*

For the successive minima problem and the shortest independent vectors problem our approach to determine short vectors outside a given subspace leads to an algorithm finding optimal solutions only for instances of Smp/Sivp respectively, where the n-th successive minimum $\lambda_n^{(\|\cdot\|)}(L)$ is bounded by $c \cdot \lambda_1^{(\|\cdot\|)}(L)$ for some constant c.

Theorem 5.0.5. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$ and $L \subseteq \mathbb{Q}^n$ be a full-dimensional lattice. Assume that the n-th successive minimum $\lambda_n^{(\|\cdot\|)}(L)$ is bounded by $c \cdot \lambda_1^{(\|\cdot\|)}(L)$ for some constant $c \in \mathbb{N}$. Then, with success probability $1 - 2^{-\Omega(n)}$, the successive minima of L can be computed using $(2^n \log_2(r))^{\mathcal{O}(1)}$ arithmetic operations, where r is an upper bound on the size of the lattice. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most $(n \cdot \log_2(r))^{\mathcal{O}(1)}$.*

Similarly, in single exponential time, we can determine the closest lattice vector to a given target vector provided that the distance of the target vector to the lattice is not too large, i.e., smaller than $c \cdot \lambda_1^{(\|\cdot\|)}(L)$ for some constant c. This variant of the closest vector problem is also called the bounded distance decoding problem (Bdd). Overall, we obtain the following results for the closest vector problem.

Theorem 5.0.6. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{Q}^n$ be a full-dimensional lattice and $t \in \operatorname{span}(L) \cap \mathbb{Q}^n$ be some target vector. Assume that there exists some constant c such that $\mu^{(\|\cdot\|)}(t, L) \leq c \cdot \lambda_1^{(\|\cdot\|)}(L)$. Then, a closest lattice vector to t can be computed using at most $(2^n \log_2(r))^{\mathcal{O}(1)}$ arithmetic operations, where r is an upper bound on the size of the lattice and the target vector. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most $(n \cdot \log_2(r))^{\mathcal{O}(1)}$.*

Further related results Based on our results, Arvind and Joglekar developed a probabilistic algorithm that solves the generalized shortest vector problem with respect to the Euclidean norm with probability exponentially close to 1. The number of arithmetic operations of their algorithm is $(2^n(1/\epsilon)^k \log_2(r))^{\mathcal{O}(1)}$, where n is the rank of the lattice, k is the dimension of the subspace and r is an upper bound on the size of the lattice and the subspace, see [AJ08]. Furthermore, they showed that a variant of their algorithm solves the generalized shortest vector problem exactly using $(2^n k^k \log_2(r))^{\mathcal{O}(1)}$ arithmetic operations. At the end of this chapter, when we have a deeper insight into the techniques used in the AKS-algorithm, we will discuss why it does not seem to be possible to generalize their approach to non-Euclidean norms.

Recently Eisenbrand, Hähnle, and Niemeier developed a probabilistic algorithm that solves the closest vector problem with respect to the ℓ_∞-norm with approximation factor $1+\epsilon$, see [EHN11]. The number of arithmetic operations of their algorithm is $(2\log_2(1/\epsilon))^{\mathcal{O}(n)} \log_2(r)^{\mathcal{O}(1)}$. The idea of their algorithm is to use our CVP-algorithm for some fixed approximation factor, e.g. for the approximation factor 2. Given a 2-approximation of the closest vector problem they use a covering of the ℓ_∞-unit ball $\bar{B}_n^{(\infty)}(0,1)$ with ellipsoids to obtain a $(1+\epsilon)$-approximation of the closest vector problem.

The main idea of the sampling procedure is to sample a large number of vectors x_i, $1 \leq i \leq N$, from a ball $\bar{B}_n^{(\|\cdot\|)}(0,\rho)$ in $\mathbb{R}^n$ for some parameter $\rho > 0$. For each vector we compute a translation $y_i \in \mathbb{R}^n$, $1 \leq i \leq N$, from the fundamental parallelepiped which translates the vector x_i to a lattice vector. So, $x_i - y_i$ is a lattice vector. One can show that if we sample enough vectors (a number single exponential in the dimension), then there exist translations y_i, y_j, $1 \leq i < j \leq N$ which are close with respect to the norm $\|\cdot\|$. In this case we have found a lattice vector of small length since $(x_i - y_i) - (x_j - y_j)$ is a lattice vector whose length is at most

$$\|(x_i - y_i) - (x_j - y_j)\| \leq \|x_i - x_j\| + \|y_i - y_j\| \leq 2\rho + \|y_i - y_j\|.$$

The translations y_i, y_j, $1 \leq i < j \leq N$, which are close together are found using a sieving procedure.

The presentation of our sampling procedure closely follows Regev's lecture notes on the Ajtai, Kumar and Sivakumar single exponential algorithm for SVP, see [Reg04], and the survey in [Eis10].

In order to almost uniformly select a vector in a ball $\bar{B}_n^{(\|\cdot\|)}(x,\rho)$ we can use the general algorithm of Dyer, Frieze, and Kannan and its improvement by Kannan, Lovász, and Simonovits, see [DFK91] and [KLS97]. This algorithm is a polynomial time algorithm that uniformly selects a vector in any well-bounded convex body given by a membership oracle. Actually, the algorithm requires that the convex body is given by a separation oracle. Grötschel, Lovász, and Schrijver show that it is possible to construct a separation oracle in polynomial time if the convex body $\mathcal{C} \subseteq \mathbb{R}^n$ is given by a membership oracle together with parameters $R, r > 0$ and a vector $c_0 \in \mathbb{R}^n$ satisfying $\bar{B}_n^{(2)}(c_0, r) \subseteq \mathcal{C} \subseteq \bar{B}_n^{(2)}(0, R)$, see [GLS93]. For a proof of the following result see [DFK91] and [KLS97].

Theorem 5.0.7. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex body given by a membership oracle together with a parameter $\gamma \geq 1$ such that $\bar{B}_n^{(2)}(0, 2^{-\gamma}) \subseteq \mathcal{C} \subseteq \bar{B}_n^{(2)}(0, 2^{\gamma})$. Then, there exists a randomized polynomial time algorithm that selects a random vector in $\mathcal{C}$ almost uniformly in the sense that its distribution is at most ϵ away from the uniform distribution in total variation distance. The number of calls to the oracle is $(n \cdot \gamma)^{\mathcal{O}(1)}$.*

The parameter $\gamma \geq 1$ is arbitrary. It can be a constant or a function of any parameter associated to the lattice.

In particular, Theorem 5.0.7 shows that for every tractable norm $\|\cdot\|$ on $\mathbb{R}^n$ we are able to efficiently select a random vector in $\bar{B}_n^{(\|\cdot\|)}(x, \rho)$ almost uniformly, where $x \in \mathbb{R}^n$ and $\alpha > 0$. For ℓ_p-norms with $1 \leq p < \infty$, there exists a simple algorithm to efficiently sample from $\bar{B}_n^{(p)}(x, \rho)$, see [GG00].

For the sake of simplicity, we will neglect all implementation details in the following, i.e., we will assume that we are able to uniformly select a vector in $\bar{B}_n^{(\|\cdot\|)}(x, \rho)$. Since we use a polynomial time algorithm to sample a vector almost uniformly, the size of each vector is at most $r^{n^{\mathcal{O}(1)}}$, where r is an upper bound on the size of the center x, the size of the radius ρ and the dimension n.

Organization This chapter is organized as follows: In Section 5.1 we show that the generalized shortest vector problem can be approximated with factor $1 + \epsilon$ for arbitrary $0 < \epsilon < 3/2$ using a variant of the AKS-sampling method. Then, we will slightly modify the sampling method and its analysis to obtain a probabilistic algorithm that solves the generalized shortest vector problem exactly, provided that there do not exist too many short lattice vectors outside the given subspace. This is done in Section 5.2. Furthermore, we will show in this section that in the case of SVP and for restricted versions of SMP, SIVP, and CVP, this assumption is always satisfied, i.e., we obtain probabilistic single exponential time algorithms that solve SVP and restricted versions of SMP, SIVP, and CVP exactly.

5.1. A sampling procedure for approximate GSVP

In this section, we present a probabilistic algorithm that solves the generalized shortest vector problem for all tractable norms with approximation factor $1 + \epsilon$ for arbitrary $0 < \epsilon < 3/2$. Before we present a detailed description of the sampling procedure, we start with some general observations.

5.1.1. Preparations

First of all, we observe that with respect to the Euclidean norm, the generalized shortest vector problem can be approximated in polynomial time with approximation factor 2^n.

Theorem 5.1.1. *The LLL-algorithm can be used to approximate the generalized shortest vector problem for the ℓ_2-norm with approximation factor 2^{n-1} in polynomial time.*

Proof. Let $L \subseteq \mathbb{Q}^n$ be a full-dimensional lattice and $M \subsetneq \operatorname{span}(L)$ be some subspace. Let $B = [b_1, \dots, b_n]$ be an LLL-reduced basis of the lattice, see Definition 4.1.9 in Chapter 4. Define

$$k := \min\{1 \leq j \leq n | b_j \in L \backslash M\},$$

that means $b_1, \dots, b_{k-1} \in M$. Since $L \neq M$, the index k is well-defined. We want to show that b_k is a 2^{n-1}-approximate solution of the generalized shortest vector problem, i.e., $\|b_k\|_2 \leq 2^{n-1}\lambda_M^{(2)}(L)$.

In the following, we consider the orthogonal projection π_k onto the orthogonal complement of $\operatorname{span}(L_{k-1})$, see Section 3.1 in Chapter 3. To recall, π_k is defined as

$$\pi_k : \mathbb{R}^n \to \mathbb{R}^n, x \mapsto \sum_{j=k}^{n} \frac{\langle x, b_j^\dagger\rangle}{\langle b_j^\dagger, b_j^\dagger\rangle} b_j^\dagger,$$

where $[b_1^\dagger, \dots, b_n^\dagger]$ is the Gram-Schmidt orthogonalization of the basis B.
Let $v \in L\backslash M$. Then we have $v = \sum_{i=1}^n v_i b_i$ with $v_i \in \mathbb{Z}$ for all $1 \leq i \leq m$. By definition of k and since $v \in L\backslash M$, there exists an index $j \geq k$ with $v_j \neq 0$. This shows that

$$\pi_k(v) = \sum_{i=k}^{n} v_i \pi_k(b_i) \neq 0,$$

i.e., we have $\pi_k(v) \in L^{(n-k+1)}\backslash\{0\}$. Since $\|v\|_2 \geq \|\pi_k(v)\|_2$, it follows that the subspace avoiding minimum of the lattice L and the subspace M is at least the minimum distance of the lattice $L^{(n-k+1)}$,

$$\lambda_M^{(2)}(L) \geq \lambda_1^{(2)}(L^{(n-k+1)}). \tag{5.1}$$

Since B is an LLL-reduced basis, the basis $[\pi_k(b_k), \dots, \pi_k(b_n)]$ of the lattice $L^{(n-k+1)}$ is also LLL-reduced, see Definition 4.1.9 in Chapter 4. From the properties of an LLL-reduced basis, we obtain

$$\|b_k^\dagger\|_2^2 \leq 2^{n-1}\lambda_1^{(2)}(L^{(n-k+1)})^2 \leq 2^{n-1}\lambda_M^{(2)}(L)^2, \tag{5.2}$$

see Theorem 4.1.11 in Chapter 4. Since $[b_1, \dots, b_n]$ is LLL-reduced, we have

$$b_k = b_k^\dagger + \sum_{j=1}^{k-1} \mu_{k,j} b_j^\dagger \text{ with } |\mu_{k,j}| \leq \frac{1}{2}$$

and

$$\|b_j^\dagger\|_2^2 \leq 2^{k-j}\|b_k^\dagger\|_2^2$$

for all $1 \leq j \leq k$. Hence, we obtain that

$$\begin{aligned}\|b_k\|_2^2 &\leq \|b_k^\dagger\|_2^2 + \frac{1}{4}\sum_{j=1}^{k}\|b_j^\dagger\|_2^2 \\ &\leq \left(1 + \frac{1}{2}\sum_{j=1}^{k-1} 2^{k-j}\right)\|b_k^\dagger\|_2^2 \\ &\leq 2^{k-1}\|b_k^\dagger\|_2^2.\end{aligned}$$

Combining this with Inequality (5.1) and Inequality (5.2), the statement follows,

$$\|b_k\|_2^2 \leq 2^{k-1}2^{n-1}\lambda_M^{(2)}(L)^2 \leq 2^{2(n-1)}\lambda_M^{(2)}(L).$$

□

There exists a generalization of the LLL-algorithm from the Euclidean norm to general norms which is due to Lovász and Scarf, see [LS92]. This algorithm is called the generalized basis reduction algorithm. Using this algorithm, we are able to approximate the generalized shortest vector problem with approximation factor 2^{2n}. This can be shown using the same techniques as in the proof of Theorem 5.1.1. Unfortunately, up to now it is not known whether the number of arithmetic operations of the generalized basis reduction algorithm is polynomial in the dimension.

If we consider a tractable norm $\|\cdot\|$ on $\mathbb{R}^n$, we can use the polynomial time approximation algorithm for the generalized shortest vector problem with respect to the Euclidean norm presented in Theorem 5.1.1 to approximate the generalized shortest vector problem with respect to the norm $\|\cdot\|$. This approximation can be used to show that we can restrict ourselves to instances of generalized shortest vector problem, where the subspace avoiding minimum of the lattice L and the subspace M satisfies $2 \leq \lambda_M^{(\|\cdot\|)}(L) < 3$.

Lemma 5.1.2. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$ and $c \in \mathbb{Z}[X]$ a polynomial satisfying $2^{-c(n)}\|x\| \leq \|x\|_2 \leq 2^{c(n)}\|x\|$ for all $x \in \mathbb{R}^n$. If there exists an algorithm $\mathcal{A}$ that for all full-dimensional lattices $L \subseteq \mathbb{Q}^n$ and all subspaces $M \subsetneq \operatorname{span}(L)$ with $2 \leq \lambda_M^{(\|\cdot\|)}(L) < 3$ solves* GSVP *with approximation factor $1+\epsilon$ using $T = T(n, r, \epsilon)$ arithmetic operations, then there exists an algorithm $\mathcal{A}'$ that solves* GSVP *for all lattices and subspaces with approximation factor $1+\epsilon$ using at most $\mathcal{O}((n + c(n)) \cdot T + n^4 \cdot \log_2(r))$ arithmetic operations, where r is an upper bound on the lattice and the subspace.*

Proof. Given a lattice $L = \mathcal{L}(B)$ and a subspace M, a vector $v \in L \backslash M$ satisfying

$$\lambda_M^{(2)}(L) \leq \|v\|_2 < 2^{n-1}\lambda_M^{(2)}(L) \tag{5.3}$$

can be computed using the LLL-algorithm, see Theorem 5.1.1. Since the norm $\|\cdot\|$ is tractable, we know that

$$2^{-c(n)}\|x\| \leq \|x\|_2 \leq 2^{c(n)}\|x\| \tag{5.4}$$

for all $x \in \mathbb{R}^n$. We set

$$\tilde{\lambda}_M(L) := \|v\|$$

as an estimate for the subspace avoiding minimum. Using the Inequalities in (5.3) and (5.4), we see that this estimate satisfies

$$\begin{aligned}\lambda_M^{(\|\cdot\|)}(L) \leq \|v\| &= \tilde{\lambda}_M(L)\\ &\leq 2^{c(n)} \cdot \|v\|_2\\ &\leq 2^{c(n)} \cdot 2^{n-1}\lambda_M^{(2)}(L)\\ &\leq 2^{2c(n)} \cdot 2^{n-1}\lambda_M^{(\|\cdot\|)}(L).\end{aligned}$$

Using the estimate $\tilde{\lambda}_M(L)$ for the subspace avoiding minimum, we want to scale the lattice such that the subspace avoiding minimum is in the range between 2 and 3. To do this, we apply algorithm $\mathcal{A}$ with the GSVP-instances (L_k, M_k), $k = 0, \ldots, 2(n + 2c(n))$, where the lattice $L_k := \mathcal{L}(B_k)$ is defined by the basis

$$B_k := \frac{1}{\tilde{\lambda}_M(L)}\left(\frac{3}{2}\right)^k B$$

and the subspace M_k is given as

$$M_k := \frac{1}{\tilde{\lambda}_M(L)}\left(\frac{3}{2}\right)^k M.$$

Let $v_0, \ldots v_{2(n+2c(n))}$ be the vectors computed by the algorithm $\mathcal{A}$. Define

$$v'_k := \tilde{\lambda}_M(L)\left(\frac{2}{3}\right)^k \cdot v_k$$

and output the shortest vector among the vectors $v'_0, \ldots, v'_{2(n+2c(n))}$ that is contained in $L \backslash M$.

First of all, we show that there exists an index $k \in \{0, \ldots, 2(n + 2c(n))\}$ such that $2 \leq \lambda_{M_k}^{(\|\cdot\|)}(L_k) < 3$. It is easy to see that a vector $v \in L \backslash M$ is a shortest vector in $L \backslash M$, i.e., $\|v\| = \lambda_M^{(\|\cdot\|)}(L)$ if and only if the vector

$$v := \frac{1}{\tilde{\lambda}_M(L)}\left(\frac{3}{2}\right)^k v \in L_k \backslash M_k$$

is a shortest vector in $L_k \backslash M_k$. This shows that

$$\lambda_{M_k}^{(\|\cdot\|)}(L_k) = \frac{1}{\tilde{\lambda}_M(L)} \left(\frac{3}{2}\right)^k \lambda_M^{(\|\cdot\|)}(L).$$

Thus the subspace avoiding minimum of a GSVP-instance (L_k, M_k) is contained in the interval $[2,3)$, if and only if

$$2 \leq \left(\frac{3}{2}\right)^k \frac{\lambda_M^{(\|\cdot\|)}(L)}{\tilde{\lambda}_M(L)} < 3.$$

That means the parameter k must satisfy

$$\frac{1 + \log_2(\tilde{\lambda}_M(L)) - \log_2(\lambda_M^{(\|\cdot\|)}(L))}{\log_2(3) - 1} \leq k < \frac{\log_2(3) + \log_2(\tilde{\lambda}_M(L)) - \log_2(\lambda_M^{(\|\cdot\|)})}{\log_2(3) - 1}. \quad (5.5)$$

The length of this interval is exactly 1, i.e., there exists an integer $k \in \mathbb{Z}$ satisfying (5.5). Since $\tilde{\lambda}_M(L) \geq \lambda_M^{(\|\cdot\|)}(L)$, the lower bound of the interval (5.5) is at least 0. Furthermore, it follows from $\tilde{\lambda}_M(L) \leq 2^{2c(n)} 2^{n-1} \lambda_M^{(\|\cdot\|)}(L)$ that the upper bound of the interval (5.5) is at most $2(2c(n)+n)$. This shows that there exists an index $k \in \{0, \ldots, 2(2c(n)+n)\}$ such that $2 \leq \lambda_{M_k}^{(\|\cdot\|)}(L_k) < 3$.

For this k the algorithm $\mathcal{A}$ computes a $(1+\epsilon)$-approximation of a shortest vector $v_k \in L_k \backslash M_k$ and the corresponding vector $v_k' := \tilde{\lambda}_M(L) \cdot (2/3)^k v_k$ is a $(1+\epsilon)$-approximation of a shortest vector in $L \backslash M$.

The number of arithmetic operations stated in the lemma follows from the number of arithmetic operations used by the LLL-algorithm, see Theorem 4.1.10 in Chapter 4. □

5.1.2. Description of the sampling procedure

In this section, we present a sampling procedure that solves the generalized shortest vector problem for all tractable norms with approximation factor $1+\epsilon$ for arbitrary $0 < \epsilon \leq \sqrt{2} - 1$. As we have seen in Lemma 5.1.2, we can assume that we are given a GSVP-instance in form of a full-dimensional lattice $L \subseteq \mathbb{Q}^n$ and some subspace $M \subsetneq \operatorname{span}(L)$ with $2 < \lambda_M^{(\|\cdot\|)}(L) < 3$.

The sieving procedure

The main part of the sampling procedure is a sieving procedure that is presented in Algorithm 1. The sieving procedure finds in any set of vectors $\{x_1, \ldots, x_N\} \subseteq \mathbb{R}^n$ inside a ball of radius R a subset J of at most $(2a+1)^n$ 'representatives' such that any vector has a representative within a distance of at most R/a. This means that the sieving procedure constructs a mapping $\sigma : \{1, \ldots, N\} \to J$ with $||x_i - x_{\sigma(i)}|| \leq R/a$ for all $i \in \{1, \ldots, N\}$. The parameter N is arbitrary. However, since we want to achieve an algorithm with single exponential running time, the sieving procedure makes sense only if $N = 2^{\mathcal{O}(n)}$. The parameter a is rational and $a > 1$. A detailed description of the sieving procedure is presented in Algorithm 1.

Algorithm 1 The sieving procedure

```
Input: $x_1, \dots, x_N \in \bar{B}_n^{(\|\cdot\|)}(0, R)$ for some parameter $R > 0$ and $a \in \mathbb{Q}$ with $a > 1$.
Output: Index set $J \subseteq \{1, \dots, N\}$ and a mapping $\sigma : \{1, \dots, N\} \to J$.
    1. Set $J \leftarrow \emptyset$.
    2. For $1 \leq j \leq N$,
       if there exists $i \in J$ with $\|x_i - x_j\| \leq R/a$, then $\sigma(j) \leftarrow i$.
       Otherwise, set $J \leftarrow J \cup \{i\}$ and $\sigma(i) \leftarrow i$.
```

The main properties of the sieving procedure are described in the following lemma.

Lemma 5.1.3. *Let $\|\cdot\|$ be an efficiently computable norm on $\mathbb{R}^n$. Let $R \in \mathbb{R}$, $R > 0$, $a \in \mathbb{Q}$ with $a > 1$. For any set of vectors $x_1, \dots, x_N \in \bar{B}_n^{(\|\cdot\|)}(0, R)$ the sieving procedure, Algorithm 1, finds a subset $J \subseteq \{1, 2, \dots, N\}$ of size of at most $(2a+1)^n$ and a mapping $\sigma : \{1, 2, \dots, N\} \longrightarrow J$ such that for all $i \in \{1, \dots, N\}$, $||x_i - x_{\sigma(i)}|| \leq R/a$. The number of arithmetic operations of the sieving procedure is $N^2 \left(n \cdot \log_2(r)\right)^{\mathcal{O}(1)}$, where r is an upper bound on the size of the vectors x_i, $1 \leq i \leq N$.*

Proof. Obviously for all $i \in \{1, \dots, N\}$, $||x_i - x_{\sigma(i)}|| \leq R/a$. We now show that $|J| \leq (2a+1)^n$. By definition of the mapping σ, the distance between any two vectors in J is greater than R/a. If we consider balls of radius $R/(2a)$ around each vector x_i, $i \in J$, then these balls are disjoint:

$$\bar{B}_n^{(\|\cdot\|)}\left(x_i, \frac{R}{2a}\right) \cap \bar{B}_n^{(\|\cdot\|)}\left(x_j, \frac{R}{2a}\right) = \emptyset \text{ for all } i, j \in J,\ i \neq j.$$

Because of $x_i \in \bar{B}_n^{(\|\cdot\|)}(0, R)$ the union of the balls $\bar{B}_n^{(\|\cdot\|)}(x_i, R/(2a))$ is contained in $\bar{B}_n^{(\|\cdot\|)}(0, (1 + 1/(2a))R)$. Therefore, the number of balls (and hence also $|J|$) is bounded by

$$\frac{\mathrm{vol}_n\left(\bar{B}_n^{(\|\cdot\|)}\left(0, (1 + \frac{1}{2a})R\right)\right)}{\mathrm{vol}_n\left((\bar{B}_n^{(\|\cdot\|)}\left(0, \frac{1}{2a}R\right)\right)} = \frac{\left(\frac{2a+1}{2a}\right)^n}{\left(\frac{1}{2a}\right)^n} = (2a+1)^n,$$

where we use Equation (2.1) from Chapter 2. The number of iterations is N. In the j-th iteration with $1 \leq j \leq N$ we consider the set J that contains at most j vectors.

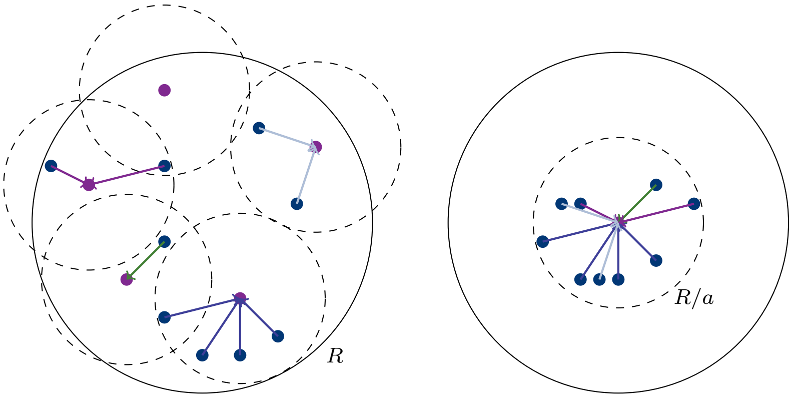

Figure 5.1.: **The effect of the sieving procedure.** Given as input a set of vectors y_i, $1 \leq i \leq N$, the sieving procedure computes for each vector y_i a representative $y_{\sigma(i)}$ with $\|y_i - y_{\sigma(i)}\| < R/a$. This is illustrated on the left. On the right, we see the vectors $y_i - y_{\sigma(i)}$, $1 \leq i \leq N$, which are contained in a $\| \cdot \|$-ball with radius R/a.

Therefore, we need to evaluate the norm roughly

$$\sum_{j=1}^{N} j = \mathcal{O}(N^2)$$

times. Since the norm $\| \cdot \|$ is efficiently computable, the number of arithmetic operations of the sieving procedure is

$$N^2 (n \cdot \log_2(r))^{\mathcal{O}(1)},$$

where r is an upper bound on the size of the vectors x_i, $1 \leq i \leq N$. □

Description of the sampling procedure

Now, we present a sampling procedure that for all tractable norms approximates the generalized shortest vector problem with approximation factor $1 + \epsilon$, $0 < \epsilon \leq 3/2$.

The algorithm chooses N vectors uniformly at random in a ball $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$ with radius $\rho > 0$. The parameter N will be defined later. For each vector x_i with $i \in \{1, \dots, N\}$ we compute the vector y_i in the fundamental parallelepiped such that $y_i - x_i$ is a lattice vector. For $x_i = \sum_{j=1}^{n} \alpha_j b_j$ with $\alpha_j \in \mathbb{Q}$ we have $y_i = \sum_{j=1}^{n} (\alpha_j - \lfloor \alpha_j \rfloor) b_j$. We apply the sieving procedure repeatedly to the vectors y_i. Using the mapping $\sigma : \{1, \dots, N\} \longrightarrow J$, for each y_i we get a representative $y_{\sigma(i)}$ with $||y_i - y_{\sigma(i)}|| < R/a$. We replace y_i with $y_i - (y_{\sigma(i)} - x_{\sigma(i)})$. This effect is illustrated in Figure 5.1.

This procedure is repeated until the distance between the lattice vectors and their representatives is small enough. Then we can show that we have found short lattice vectors.

Algorithm 2 The sampling procedure

```
Input:
  • A lattice basis $B = [b_1, \dots, b_m]$ of a lattice $L \subseteq \mathbb{R}^n$,
  • a subspace $M \subsetneq \text{span}(L)$, and
  • parameters $0 < \delta < \sqrt{2} - 1$ and $\rho \geq 1/2$.

Used subroutine: Sieving procedure.

Output: A vector $v \in L \backslash M$ or "failure".

  1.  a) Set $R_0 \leftarrow m \cdot \max\{\|b_i\| \,|1 \leq i \leq m\}$.
      b) Choose $N$ vectors $x_1, \dots, x_N$ uniformly in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$.
      c) Compute $y_i \in \mathcal{P}(B)$ with $y_i \equiv x_i \mod L$ for $i = 1, \dots, N$.
      d) Set $\mathcal{Z} \leftarrow \{(x_1, y_1), \dots, (x_N, y_N)\}$.
      e) Set $R \leftarrow R_0$ and $a \leftarrow 1 + 2/\delta$.
  2.  While $R > (1 + \delta)\rho$,
      a) apply the sieving procedure to the set $\{y_i|(x_i, y_i) \in \mathcal{Z}\}$ with the parameters
         $a = 1 + 2/\delta$ and $R$. The result is a set $J$ and a mapping $\sigma$.
      b) Remove all pairs $(x_i, y_i)$ with $i \in J$ from $\mathcal{Z}$.
      c) Replace each remaining pair $(x_i, y_i) \in \mathcal{Z}$ with $(x_i, y_i - (y_{\sigma(i)} - x_{\sigma(i)}))$.
      d) Set $R \leftarrow R/a + \rho$.
  3.  Set $\mathcal{S} := \{y_i - x_i|(x_i, y_i) \in \mathcal{Z}\}$.
      Output a shortest vector $v \in \mathcal{S}$ with $v \notin M$ if such a vector exists. Otherwise,
      the output is "failure".
```

A detailed description of the algorithm is presented in Algorithm 2.

We use parameters δ and ρ satisfying

$$\begin{aligned} &0 < \delta \leq \sqrt{2} - 1 \text{ and} \\ &\rho \geq 1/2. \end{aligned}$$

The required upper bound for the parameter δ is needed to show that the sampling procedure solves the generalized shortest vector problem with probability exponentially close to 1. The lower bound on the radius ρ is required to give an upper bound on the number of arithmetic operations of the algorithm and also influences the success probability of the algorithm.

We have $\sigma(i) = i$ for each pair $(x_i, y_i) \in \mathcal{Z}$ with $i \in J$ and therefore $y_i - (y_{\sigma(i)} - x_{\sigma(i)}) - x_i = 0$. By removing in step 2b) each pair (x_i, y_i) with $i \in J$ from $\mathcal{Z}$ we avoid redundant elements.

Now, we analyze the sampling procedure and state its main properties. Furthermore, the main part of the sampling procedure is the application of the sieving procedure. In each application of the sieving procedure we remove all pairs (x_i, y_i) with $i \in J$ from $\mathcal{Z}$. To derive results about the success probability of the sampling procedure, we need to guarantee that at the end of the sampling procedure the set $\mathcal{Z}$ contains sufficiently many vectors. Hence, we are interested in an upper bound on the number of pairs which are removed during the sampling procedure.

We start with the analysis of the output and show that the sampling procedure solves the generalized shortest vector problem correctly if it outputs a vector v and not "failure".

Lemma 5.1.4. *Given a lattice basis $B \in \mathbb{Q}^{n \times n}$ and a subspace $M \subsetneq \operatorname{span}(B)$ with parameters ρ and δ chosen as above, the sampling procedure, Algorithm 2, outputs a vector $v \in L \backslash M$ of length at most $(2+\delta) \cdot \rho$, when it is successful.*

Proof. During the sampling procedure, two invariants are maintained:

1. For all $(x_i, y_i) \in \mathcal{Z}$, $y_i - x_i \in \mathcal{L}(B)$.
2. For all $(x_i, y_i) \in \mathcal{Z}$, $\|y_i\| \leq R$.

Let us consider the first invariant. The algorithm chooses N vectors $x_1, \ldots, x_N$ in $\bar{B}_n^{(\|\cdot\|)}(0, r)$ and computes y_i with $y_i \equiv x_i \mod \mathcal{L}(B)$ for $i \in \{1, \ldots, N\}$. That means, $y_i - x_i \in \mathcal{L}(B)$. During the while-loop in step 2 of the sampling procedure we only subtract from y_i vectors of the form $y_j - x_j$ that are themselves lattice vectors.

Next, we consider the second invariant. At the start of the while-loop we have $y_i \in \mathcal{P}(B)$. Hence, for all $i \in \{1, \ldots, N\}$ the length of y_i is bounded by

$$\|y_i\| \leq \sum_{j=1}^{n} \|b_i\| \leq n \cdot \max\{\|b_j\| \mid 1 \leq j \leq m\} = R_0 = R.$$

This property is maintained during each iteration of the while-loop since the distance between every vector y_i and its corresponding representative is at most R/a,

$$\left\|y_i - (y_{\sigma(i)} - x_{\sigma(i)})\right\| \leq \left\|y_i - y_{\sigma(i)}\right\| + \left\|x_{\sigma(i)}\right\| \leq \frac{R}{a} + \left\|x_{\sigma(i)}\right\|,$$

see Lemma 5.1.3. Since $x_{\sigma(i)} \in \bar{B}_n^{(\|\cdot\|)}(0, \rho)$, it follows that

$$\left\|y_i - (y_{\sigma(i)} - x_{\sigma(i)})\right\| < \frac{R}{a} + \rho \leq R.$$

The last inequality is based on the fact that at the end of each iteration of the while-loop R is replaced by $R/a + \rho$.

If $R \leq (1+\delta)\rho$, the while-loop terminates. By the two invariants, each remaining pair $(x_i, y_i) \in \mathcal{Z}$ satisfies $y_i - x_i \in \mathcal{L}(B)$ and

$$\|y_i - x_i\| \leq \|y_i\| + \|x_i\| \leq (1+\delta)\rho + \rho = (2+\delta)\rho.$$

□

Now, we consider the number of arithmetic operations used by the sampling procedure that is mainly influenced by the number of iterations of the while-loop in step 2 of the sampling procedure. Here we use that the radius ρ is at least 1/2.

Claim 5.1.5. *Given a lattice basis $B \in \mathbb{Q}^{n \times n}$ and a subspace $M \subsetneq \operatorname{span}(B)$ together with the parameters $0 < \delta < \sqrt{2} - 1$ and $\rho \geq 1/2$, the number of iterations of the while-loop of the sampling procedure is at most*

$$2\log_2\left(1 + \frac{2}{\delta}\right) \cdot \left(\log_2(R_0) + \log_2\left(1 + \frac{2}{\delta}\right)\right),$$

where $R_0 = m \cdot \max\{\|b_i\| \mid 1 \leq i \leq m\}$ is an upper bound of the input size.

Proof. The parameter R is initialized as R_0. After i steps of the while-loop the parameter R is

$$\frac{R_0}{a^i} + \rho \sum_{j=0}^{i-1} a^{-j}.$$

The iteration terminates if $R \leq (1+\delta)\rho$. Since $a \neq 1$, we can use the geometric series

$$\frac{R_0}{a^i} + \rho \cdot \sum_{j=1}^{i-1} a^{-j} \leq \frac{R_0}{a^i} + \rho \cdot \frac{a}{a-1}$$

to see that the iteration terminates if

$$\frac{R_0}{a^i} + \rho \frac{a}{a-1} \leq (1+\delta)\rho.$$

We obtain that

$$i \geq \log_2 a \cdot (\log_2 R_0 + \log_2(a-1) - \log_2((\delta(a-1)-1)\rho)).$$

Since $a = 1 + 2/\delta$, the number of iterations in step 2 is at most

$$\begin{aligned} & \log_2(a) \cdot (\log_2(R_0) + \log_2(a-1) - \log_2((\delta(a-1)-1)\rho)) \\ \leq\ & \log_2(a)\,(\log_2(R_0) + \log_2(a) - \log_2(\rho)). \end{aligned}$$

Since $\rho \geq 1/2$, we obtain that this is at most

$$\begin{aligned} & \log_2(a) \cdot (\log_2(R_0) + \log_2(a) + 1) \\ \leq\ & 2\log_2(a) \cdot (\log_2(R_0) + \log_2(a)) \\ =\ & 2\log_2(1 + \frac{2}{\delta}) \cdot (\log_2(R_0) + \log_2(1 + \frac{2}{\delta})). \end{aligned}$$

□

Using this bound for the number of iterations, we can analyze the number of arithmetic operations of the sampling procedure. Furthermore, we are able to give an upper bound on the number of pairs which are removed from the set $\mathcal{Z}$ during the sampling procedure.

Lemma 5.1.6. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. Given a lattice basis $B \in \mathbb{Q}^{n\times n}$ and a subspace $M \subsetneq \operatorname{span}(\mathcal{L}(B))$ with the parameters ρ and δ satisfying $0 < \delta < \sqrt{2}-1$ and $\rho \geq 1/2$, the sampling procedure, Algorithm 2, satisfies the following properties:*

- *The number of arithmetic operations of the sampling procedure is bounded by*

$$\left(\log_2\left(1+\frac{2}{\delta}\right)\cdot\log_2(r)\cdot n\cdot N\right)^{\mathcal{O}(1)},$$

  where N is the number of vectors chosen in the sampling procedure and r is an upper bound on the size of the lattice basis B and the subspace M.
  The representation size of each number computed by the algorithm is at most

$$\left(n\cdot\log_2\left(1+\frac{2}{\delta}\right)\log_2(r)\right)^{\mathcal{O}(1)}.$$

- *We remove at most*

$$z(R_0,\delta) := \left(\log_2(R_0)+\log_2\left(1+\frac{2}{\delta}\right)\right)\left(2\left(1+\frac{2}{\delta}\right)+1\right)^{n+1} \tag{5.6}$$

  pairs from the set $\mathcal{Z}$.

Proof. The number of iterations of the while-loop dominates the number of arithmetic operations used by the sampling procedure and is bounded by

$$2\log_2\left(1+\frac{2}{\delta}\right)\cdot\left(\log_2(R_0)+\log_2\left(1+\frac{2}{\delta}\right)\right), \tag{5.7}$$

as we have seen in Claim 5.1.5. This term is bounded by

$$4\log_2\left(1+\frac{2}{\delta}\right)^2\cdot\log_2(R_0) \leq \left(m\log_2\left(1+\frac{2}{\delta}\right)\log_2(r)\right)^{\mathcal{O}(1)}, \tag{5.8}$$

where r is an upper bound on the size of the basis B and the subspace M.
In each iteration, we apply the sieving procedure to the set $\{y_i | (x_i, y_i) \in \mathcal{Z}\}$. Since the size of the vectors x_i sampled from $\bar{B}_n^{(\|\cdot\|)}(0,\rho)$ is at most $r^{n^{\mathcal{O}(1)}}$, it is easy to see that the size of the translations y_i is also $r^{n^{\mathcal{O}(1)}}$, see Theorem 5.0.7.
In each iteration, we change the vectors y_i by adding or subtracting two other vectors of size of at most $r^{n^{\mathcal{O}(1)}}$. Since the number of iterations is at most $(m\cdot\log_2(1+2/\delta)\cdot\log_2(r))^{\mathcal{O}(1)}$, see (5.8), the size of the vector y_i after step 2 of the sampling procedure is at most

$$r^{(n\cdot\log_2(1+2/\delta)\cdot\log_2(r))^{\mathcal{O}(1)}}.$$

Hence, in each iteration step of the while-loop, the sieving procedure is applied to vectors with this size and it follows that the number of arithmetic operations is at most

$$N^2 \left(n \cdot \log_2(r)\right)^{\mathcal{O}(1)}.$$

Combining this with the upper bound for the number of iterations of the while-loop given in (5.8), we obtain that the number of arithmetic operations of the sampling procedure is bounded by

$$\left(\log_2\left(1+\frac{2}{\delta}\right)\log_2(r)\cdot n\cdot N\right)^{\mathcal{O}(1)}.$$

Since the sieving procedure is executed at most $2\log_2(1+2/\delta)\left(\log_2(R_0)+\log_2(1+2/\delta)\right)$ times, see (5.7), and we find a set of size of at most $(2a+1)^n = (2(1+2/\delta)+1)^n$ in each application of the sieving procedure, see Lemma 5.1.3, we remove at most

$$2\log_2\left(1+\frac{2}{\delta}\right)\left(\log_2(R_0)+\log_2\left(1+\frac{2}{\delta}\right)\right)\cdot\left(2\left(1+\frac{2}{\delta}\right)+1\right)^n$$

pairs from $\mathcal{Z}$. □

Combining Lemma 5.1.4 and Lemma 5.1.6 with the right choice of parameters, we see that the sampling procedure computes a set of lattice vectors whose length is at most $(1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ for arbitrary $0<\epsilon\leq 3/2$.

Theorem 5.1.7. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. For every $0<\epsilon\leq 3/2$ there exists a $\delta>0$ such that the following holds: Given a full-dimensional lattice $L\subseteq\mathbb{Q}^n$ and a parameter ρ satisfying*

$$\frac{1}{2}\leq\rho\leq\frac{1}{2}(1+\delta)^2\lambda_M^{(\|\cdot\|)}(L),$$

the sampling procedure, Algorithm 2, computes a set $\mathcal{S}$ of vectors from $L\cap\bar{B}_n^{(\|\cdot\|)}(0,(1+\epsilon)\lambda_M^{(\|\cdot\|)}(L))$. The number of arithmetic operations of the sampling procedure is

$$\left(\log_2\left(2+\frac{1}{\epsilon}\right)\log_2(r)\cdot n\cdot N\right)^{\mathcal{O}(1)},$$

where N is the number of vectors chosen in the sampling procedure and r is an upper bound on the size of the lattice L and the subspace M. The representation size of each number computed by the sampling procedure is at most

$$\left(n\cdot\log_2\left(2+\frac{1}{\epsilon}\right)\cdot\log_2(r)\right)^{\mathcal{O}(1)}.$$

Proof. If we choose $\delta = \epsilon/4$, it follows from $\epsilon \le 3/2$ that $\delta < \sqrt{2} - 1$. The sampling procedure computes a set of pairs $(x, y) \in \mathcal{Z}$ each satisfying $\|y - x\| \le (2 + \delta)\rho$ and $y - x \in L$, see Lemma 5.1.6. Since the parameter ρ satisfies $\rho \le (1/2) \cdot (1 + \delta)^2 \lambda_M^{(\|\cdot\|)}(L)$ we get

$$\|y - x\| \le (2 + \delta)\frac{1}{2}(1 + \delta)^2 \lambda_M^{(\|\cdot\|)}(L) = \left(1 + \frac{\delta}{2}(5 + 4\delta + \delta^2)\right) \lambda_M^{(\|\cdot\|)}(L).$$

Since $\delta = \epsilon/4$, it follows that

$$\|y - x\| \le (1 + \epsilon)\lambda_M^{(\|\cdot\|)}(L).$$

Using Lemma 5.1.6, the number of arithmetic operations of the sampling procedure is

$$\left(\log_2\left(1 + \frac{2}{\delta}\right) \cdot \log_2(r) \cdot n \cdot N\right)^{\mathcal{O}(1)} = \left(\log_2\left(2 + \frac{1}{\epsilon}\right) \cdot \log_2(r) \cdot n \cdot N\right)^{\mathcal{O}(1)}$$

and each number computed by the algorithm has representation size of at most

$$\left(n \cdot \log_2\left(1 + \frac{2}{\delta}\right) \cdot \log_2(r)\right)^{\mathcal{O}(1)} = \left(n \cdot \log_2\left(2 + \frac{1}{\epsilon}\right) \cdot \log_2(r)\right)^{\mathcal{O}(1)}.$$

□

5.1.3. Analysis of the sampling procedure using a modified sampling procedure

The sampling procedure computes a set of lattice vectors whose length is at most $(1 + \epsilon)\lambda_M^{(\|\cdot\|)}(L)$. So far, we have not excluded the case that all vectors are contained in the subspace M.

We need to show that the sampling procedure computes vectors in $L \backslash M$. For this, we use the randomization in the algorithm. We change our point of view and consider a modified sampling procedure that behaves exactly like the sampling procedure presented in Algorithm 2. We are able to show that the modified sampling procedure computes a vector $v \in L \backslash M$ with success probability $1 - 2^{-\Omega(n)}$. Hence, the same is true for the sampling procedure.

We consider a lattice vector $u \in L \backslash M$ with $\|u\| = \lambda_M^{(\|\cdot\|)}(L)$ and define the sets

$$C_1 := \bar{B}_n^{(\|\cdot\|)}(0, \rho) \cap \bar{B}_n^{(\|\cdot\|)}(u, \rho) \text{ and } C_2 := \bar{B}_n^{(\|\cdot\|)}(0, \rho) \cap \bar{B}_n^{(\|\cdot\|)}(-u, \rho).$$

If the parameter ρ satisfies

$$\frac{1}{2}(1 + \delta)\lambda_M^{(\|\cdot\|)}(L) \le \rho \le \frac{1}{2}(1 + \delta)^2 \lambda_M^{(\|\cdot\|)}(L) \tag{5.9}$$

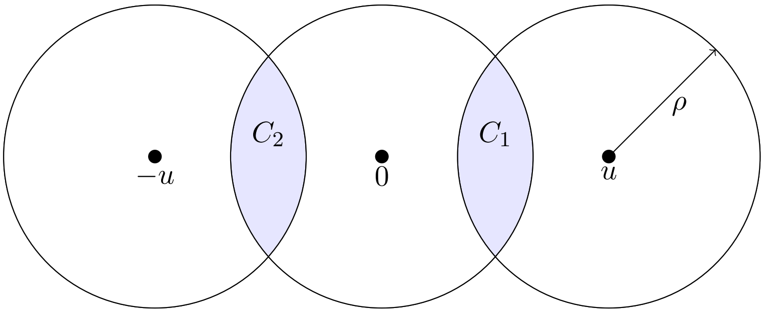

Figure 5.2.: **The sets $\mathcal{C}_1$ and $\mathcal{C}_2$.** We consider balls with radius ρ around the vectors $-u$, 0, and u. If the radius ρ is less than $\|u\|$, the intersections C_1 and C_2 are disjoint. If the radius r is greater than $\|u\|/2$, the intersections C_1 and C_2 are non-empty.

for a $0 < \delta < \sqrt{2} - 1$, we have

$$\rho > \frac{1}{2}\lambda_M^{(\|\cdot\|)}(L)$$

and

$$\rho \leq \frac{1}{2}(1+\delta)^2\lambda_M^{(\|\cdot\|)}(L) < \lambda_M^{(\|\cdot\|)}(L).$$

Therefore the sets C_1 and C_2 are non-empty and disjoint. The form of the sets with respect to the Euclidean norm is shown in Figure 5.2.

We define a mapping $\tau_u : \bar{B}_n^{(\|\cdot\|)}(0,\rho) \longrightarrow \mathbb{R}^n$ depending on the lattice vector u.

$$\tau_u(x) = \begin{cases} x + u & , \ x \in C_2 \\ x - u & , \ x \in C_1 \\ x & , \ \text{otherwise} \end{cases} . \tag{5.10}$$

Claim 5.1.8. *Let $L \subseteq \mathbb{R}^n$ be a lattice and $M \subsetneq \operatorname{span}(L)$. For $0 < \delta < \sqrt{2} - 1$ let $\rho > 0$ be a parameter satisfying (5.9). Then the mapping τ_u defined as in (5.10) is a bijective mapping*

$$\tau_u : \bar{B}_n^{(\|\cdot\|)}(0,\rho) \to \bar{B}_n^{(\|\cdot\|)}(0,\rho)$$

which maps C_1 to C_2, C_2 to C_1, and $\bar{B}_n^{(\|\cdot\|)}(0,\rho)\backslash(C_1 \cup C_2)$ to itself. Particularly, we have $\|\tau_u(x)\| \leq \rho$ for all $x \in \bar{B}_n^{(\|\cdot\|)}(0,\rho)$.

Proof. The statement follows directly by the definition of the sets C_1 and C_2. For $x \in C_1 = \bar{B}_n^{(\|\cdot\|)}(0,\rho) \cap \bar{B}_n^{(\|\cdot\|)}(u,\rho)$ we have

$$\begin{aligned} &\|\tau_u(x)\| = \|x - u\| \leq \rho \text{ and} \\ &\|\tau_u(x) + u\| = \|x\| \leq \rho, \end{aligned}$$

that means $\tau_u(x) \in C_2$. Analogously, we see that $\tau_u(x) \in C_1$ for all $x \in C_2$ and that $\tau_u(x) \in \bar{B}_n^{(\|\cdot\|)}(0, \rho) \backslash (C_1 \cup C_2)$ for all $x \in \bar{B}_n^{(\|\cdot\|)}(0, \rho) \backslash (C_1 \cup C_2)$. Obviously, τ_u is a bijective mapping. □

Using the mapping τ_u we define the modified sampling procedure which is presented in Algorithm 3. The modified sampling procedure applies the sieving procedure in the same way as the original sampling procedure. The result of this sieving procedure is an index set J. In contrast to the original sampling procedure, for each pair (x_i, y_i) with $i \in J$, the modified sampling procedure replaces the vector x_i with probability $1/2$ by the vector $\tau_u(x_i)$. Furthermore, after the termination of the while-loop for each remaining pair $(x, y) \in \mathcal{Z}$, the modified sampling procedure replaces the vector x with probability $1/2$ by the vector $\tau_u(x)$.

The modified sampling procedure is only used for the analysis. Hence, we do not worry about its running time and the fact that it uses the unknown vector u. Since τ_u maps $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$ to $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$, we have $\|\tau_u(x)\| \leq \rho$ for all $x \in \bar{B}_n^{(\|\cdot\|)}(0, \rho)$. Thus, analogously to Lemma 5.1.7, we can see that the modified sampling procedure returns vectors in $L \cap \bar{B}_n^{(\|\cdot\|)}(0, (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L))$.

The sampling procedure presented in Algorithm 2 and the modified sampling procedure presented in Algorithm 3 return vectors distributed according to certain distributions. We call these the *output distributions* generated by the sampling procedure and the modified sampling procedure, respectively.

Theorem 5.1.9. *The sampling procedure, Algorithm 2, and the modified sampling procedure, Algorithm 3, generate the same output distribution.*

Proof. First, we consider the following modification in step 1b) of the sampling procedure presented in Algorithm 2. After choosing the vectors x_i we decide for each x_i uniformly at random whether to keep x_i or to replace it with $\tau_u(x_i)$. This does not change the distribution on the vectors x_i. Hence, this modification does not change the output distribution of the sampling procedure.

Next, we observe that we can postpone the decision of replacing x_i to the first time in which it has an effect on the algorithm. We observe that $u \in L$ implies

$$y_i \equiv x_i \equiv \tau_u(x_i) \mod L, i = 1, \ldots, N.$$

Hence, if we decide for each x_i whether to replace it with $\tau_u(x_i)$ at the end of step 1 rather than in step 1b), this does not change the output distribution.

But if, without changing the output distribution, we can choose for each x_i whether to keep it or to replace it with $\tau_u(x_i)$ at the end of step 1, then making that decision for each x_i prior to the first time it is used in step 2 will also not change the output distribution. Furthermore, for each vector x_i not used at all in step 2 we can choose whether to keep

Algorithm 3 The modified sampling procedure

```
Input:
    • A lattice basis $B = [b_1, \ldots, b_n]$ of a lattice $L$,
    • a subspace $M \subsetneq \mathrm{span}(L)$, and
    • parameters $0 < \delta < \sqrt{2} - 1$ and $\rho \geq 1/2$.

Used subroutine: Sieving procedure.

Output: A vector $v \in L \backslash M$ or "failure".

1.  a) Set $R_0 \leftarrow n \cdot \max\{\|b_i\| | 1 \leq i \leq n\}$.
    b) Choose $N$ vectors $x_1, \ldots, x_N$ uniformly in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$.
    c) Compute $y_i \in \mathcal{P}(B)$ with $y_i \equiv x_i \mod L$ for $i = 1, \ldots, N$.
    d) Set $\mathcal{Z} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$.
    e) Set $R \leftarrow R_0$ and $a \leftarrow 1 + 2/\delta$.

2.  While $R > (1 + \delta)\rho$,
    a) apply the sieving procedure to $\{y_i | (x_i, y_i) \in \mathcal{Z}\}$ with the parameters $a = 1 + 2/\delta$ and $R$. The result is a set $J$ and a mapping $\sigma$.
    b) Remove from $\mathcal{Z}$ all pairs $(x_i, y_i)$ with $i \in J$.
    c) For each pair $(x_i, y_i)$, $i \in J$, replace $x_i$ with $\tau_u(x_i)$ with probability $\frac{1}{2}$.
    d) Replace each remaining pair $(x_i, y_i) \in \mathcal{Z}$ with $(x_i, y_i - (y_{\sigma(i)} - x_{\sigma(i)}))$.
    e) Set $R \leftarrow R/a + \rho$.

3.  For each pair $(x_i, y_i) \in \mathcal{Z}$ replace $x_i$ with $\tau_u(x_i)$ with probability $\frac{1}{2}$.

4.  Set $\mathcal{S} := \{y_i - x_i | (x_i, y_i) \in \mathcal{Z}\}$.
    Output a shortest vector $v \in \mathcal{S}$ with $v \notin M$ if such a vector exists. Otherwise, the output is "failure".
```

it or replace it with $\tau_u(x_i)$ at the end of step 2. But this is exactly the modification leading from the sampling procedure presented in Algorithm 2 to the modified sampling procedure presented in Algorithm 3. □

Mathematically, this proof is not correct. Since we consider a continuous probability distribution on $\bar{B}_n^{(\|\cdot\|)}(0,\rho)$, the probability of a finite vector $x \in \bar{B}_n^{(\|\cdot\|)}(0,\rho)$ is 0. Hence, we cannot argue that a vector $x \in \bar{B}_n^{(\|\cdot\|)}(0,\rho)$ is chosen with the same probability as the vector $\tau_u(x) \in \bar{B}_n^{(\|\cdot\|)}(0,\rho)$. Nevertheless, the statement in Theorem 5.1.9 is correct. But to prove it correctly, we need to consider small balls around the vectors x and $\tau_u(x)$ and we have to argue that they have the same volume. For the sake of simplicity and for a better understanding we omit this proof here.

For further analysis, we need the probability, that a vector x, which is chosen uniformly in $\bar{B}_n^{(\|\cdot\|)}(0,\rho)$, is contained in $C_1 \cup C_2$.

Lemma 5.1.10. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$, let $u \in \mathbb{R}^n$ be a vector and $\zeta > 0$. Define $C := \bar{B}_n^{(\|\cdot\|)}\left(0,(1/2)(1+\zeta)\|u\|\right) \cap \bar{B}_n^{(\|\cdot\|)}\left(u, 1/2(1+\zeta)\|u\|\right)$. Then*

$$\frac{\mathrm{vol}_n(C)}{\mathrm{vol}_n\left(B_n^{(\|\cdot\|)}\left(0,\frac{1}{2}(1+\zeta)\|u\|\right)\right)} \geq \left(\frac{\zeta}{1+\zeta}\right)^n.$$

Proof. Since $(1/2)(1+\zeta)\|u\| > (1/2)\cdot\|u\|$, the intersection C is non-empty. The intersection C contains a ball with radius $(1/2)\zeta\|u\|$ centered around $u/2$, since for all $s \in \bar{B}_n^{(\|\cdot\|)}(u/2,(1/2)\zeta\|u\|)$ it holds that

$$\|s\| \leq \left\|s - \frac{u}{2}\right\| + \left\|\frac{u}{2}\right\| \leq \frac{1}{2}\zeta\|u\| + \frac{1}{2}\|u\| = \frac{1}{2}(1+\zeta)\|u\|$$

and

$$\|s-u\| \leq \left\|s - \frac{u}{2}\right\| + \left\|\frac{u}{2}\right\| = \frac{1}{2}(1+\zeta)\|u\|.$$

We get that

$$\mathrm{vol}_n(C) \geq \mathrm{vol}_n\left(\bar{B}_n^{(\|\cdot\|)}\left(0,\frac{1}{2}\zeta\|u\|\right)\right).$$

Using Equation (2.1) in Chapter 2, we obtain

$$\frac{\mathrm{vol}_n(C)}{\mathrm{vol}_n\left(B_n^{(\|\cdot\|)}(0,\frac{1}{2}(1+\zeta)\|u\|)\right)} \geq \frac{\left(\frac{1}{2}\zeta\cdot\|u\|\right)^n \mathrm{vol}_n\left(B_n^{(\|\cdot\|)}(\frac{u}{2},1)\right)}{\left(\frac{1}{2}(1+\zeta)\|u\|\right)^n \mathrm{vol}_n\left(B_n^{(\|\cdot\|)}(0,1)\right)} = \left(\frac{\zeta}{1+\zeta}\right)^n.$$

□

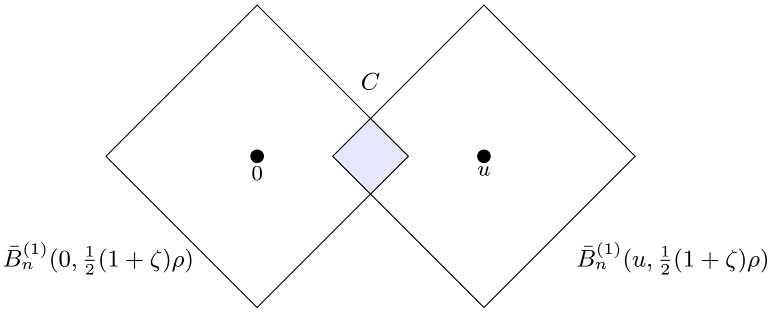
C
0
u
$\bar{B}_n^{(1)}(0, \frac{1}{2}(1+\zeta)\rho)$
$\bar{B}_n^{(1)}(u, \frac{1}{2}(1+\zeta)\rho)$

Figure 5.3.: **Volume of the intersection of two ℓ_1-balls.** The intersection $\bar{B}_n^{(1)}(0, \frac{1}{2}(1+\zeta)\rho) \cap \bar{B}_n^{(1)}(u, \frac{1}{2}(1+\zeta)\rho)$ is exactly the ℓ_1-ball with radius $(1/2)\zeta\|u\|_1$ centered at $u/2$.

For general norms, the bound given in this lemma is tight. Consider the vector $u = (u_1, 0, \dots, 0) \in \mathbb{R}^n$ with respect to the ℓ_1-norm. Then, the intersection $\mathcal{C}$ is exactly the ℓ_1-ball $\bar{B}_n^{(1)}(u/2, (1/2)\zeta|u_1|)$, see Figure 5.3. For the Euclidean norm one can achieve a slightly better bound by looking at a $(n-1)$-dimensional cylinder centered at the vector $u/2$, see [GG00] and [Reg04].

The sampling procedure and the modified sampling procedure choose N vectors uniformly at random in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$. We are interested in the number of vectors which are contained in $C_1 \cup C_2$.

Lemma 5.1.11. *Let $N \in \mathbb{N}$. By q, denote the probability that a random vector in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$ is contained in $C_1 \cup C_2$. If N vectors $x_1, \dots, x_N$ are chosen uniformly at random in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$, then with probability larger than $1 - 4/(N \cdot q)$ there are at least $(q \cdot N)/2$ vectors $x_i \in \{x_1, \dots, x_N\}$ with the property $x_i \in C_1 \cup C_2$.*

Proof. Let X be the number of vectors which are contained in $C_1 \cup C_2$. The expected number of vectors from $C_1 \cup C_2$ is $q \cdot N$ with variance $N \cdot q \cdot (1-q) < N \cdot q$. Using Chebyshev's inequality, see Theorem A.0.1 in the Appendix, we get

$$P\left(|X - \underbrace{E(X)}_{=q\cdot N}| \geq \underbrace{\frac{q \cdot N}{2}}_{=:\epsilon}\right) \leq \frac{\mathrm{Var}(X)}{\epsilon^2} < \frac{N \cdot q}{\frac{1}{4}(N \cdot q)^2} = \frac{4}{N \cdot q}.$$

Therefore,

$$P\left(|X| \leq \frac{q \cdot N}{2}\right) \leq \frac{4}{N \cdot q}.$$

□

For further analysis only pairs (x, y) with $x \in C_1 \cup C_2$ are of interest because only for them the mapping τ_u is not the identity. The next lemma shows how many vectors N one has to choose at the beginning of the sampling procedure so that at the end of step 2 of the sampling procedure the set $\mathcal{Z}$ contains sufficiently many pairs (x, y) satisfying the property that $x \in C_1 \cup C_2$.

Lemma 5.1.12. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. We consider the sampling procedure, Algorithm 2, respectively the modified sampling procedure, Algorithm 3, with input of a full-dimensional lattice $L \subseteq \mathbb{Q}^n$, a subspace $M \subsetneq \operatorname{span}(L)$ and a parameter ρ satisfying*

$$\rho \geq \frac{1}{2}(1+\delta)\lambda_M^{(\|\cdot\|)}(L)$$

for arbitrary $0 < \delta \leq 1/2$. Furthermore, assume that in the first step of the sampling procedure or the modified sampling procedure the number of vectors chosen is

$$N = \left(\frac{1+\delta}{\delta}\right)^n 2\left(\nu + z(R_0, \delta)\right),$$

where $z(R_0, \delta)$ is defined as in (5.6) and $\nu \in \mathbb{N}$. Then at the end of step 2 of the sampling procedure or the modified sampling procedure, the set $\mathcal{Z}$ contains with probability $1 - 2/\nu$ at least ν pairs (x, y) with the property $x \in C_1 \cup C_2$.

The proof combines Lemma 5.1.10, where we determined the probability that a vector x which is chosen uniformly at random in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$ is contained in $C_1 \cup C_2$, with Lemma 5.1.11, where we determined the number of vectors which are contained in $C_1 \cup C_2$, if we choose N vectors at the beginning of the sampling procedure. Additionally, we have to consider the number of pairs which are removed from the set $\mathcal{Z}$.

Proof. If $\rho \geq (1/2)(1+\delta)\lambda_M^{(\|\cdot\|)}(L)$, we have $\rho = (1/2)(1+\zeta)\lambda_M^{(\|\cdot\|)}(L)$ for some $\zeta \geq \delta$. Using Lemma 5.1.10 with $\|u\| = \lambda_M^{(\|\cdot\|)}(L)$ and ζ, we obtain that

$$\begin{aligned} \frac{\operatorname{vol}_n(C_1)}{\operatorname{vol}_n\left(\bar{B}_n^{(\|\cdot\|)}(0,\rho)\right)} &\geq \left(\frac{\zeta}{1+\zeta}\right)^n \\ &\geq \left(\frac{\delta}{1+\delta}\right)^n, \end{aligned} \tag{5.11}$$

where the last inequality follows from $\zeta \geq \delta$. It follows from (5.11) that for $i \in \{1, \ldots, N\}$ we have $x_i \in C_1 \cup C_2$ with probability at least

$$q := \left(\frac{\delta}{1+\delta}\right)^n.$$

Using this in combination with Lemma 5.1.11, the set $\{x_1, \ldots, x_N\}$ contains with probability

$$1 - \frac{4}{N \cdot q} = 1 - \frac{2}{\nu + z(R_0, \delta)} > 1 - \frac{2}{\nu}$$

at least $(q \cdot N)/2$ vectors from $C_1 \cup C_2$. With Lemma 5.1.6, we remove at most $z(R_0, \delta)$ pairs from $\mathcal{Z}$. Therefore, at the end of the algorithm the set $\mathcal{Z}$ contains with probability larger than $1 - 2/\nu$ at least

$$\frac{1}{2} q \cdot N - z(R_0, \delta) = \nu + z(R_0, \delta) - z(R_0, \delta) = \nu$$

pairs (x, y) with the property $x \in C_1 \cup C_2$. □

Using this result, we are able to show that the modified sampling procedure computes a lattice vector which is not contained in the subspace M with probability exponentially close to 1.

Theorem 5.1.13. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. For every $0 < \epsilon \leq 3/2$ there exists a $\delta > 0$ such that the following holds: Given a lattice $L \subseteq \mathbb{Q}^n$ and a subspace $M \subsetneq \operatorname{span}(L)$ satisfying $\lambda_M^{(\|\cdot\|)}(L) \geq 2$ and a parameter ρ satisfying Equation (5.9), i.e.,*

$$\frac{1}{2}(1+\delta)\lambda_M^{(\|\cdot\|)}(L) \leq \rho \leq \frac{1}{2}(1+\delta)^2 \lambda_M^{(\|\cdot\|)}(L),$$

then the modified sampling procedure, Algorithm 3, computes a vector $v \in L \backslash M$ with probability $1 - 2^{-\Omega(n)}$.

Proof. We apply the sampling procedure with the same parameters as in Theorem 5.1.7, i.e., we choose

$$\begin{aligned} \delta &= \epsilon/4 \\ N &= ((1+\delta)/\delta)^n \, 2 \left(2^n + z(R_0, \delta)\right). \end{aligned}$$

Since $\lambda_M^{(\|\cdot\|)}(L) \geq 2$, we have $\rho \geq 1/2$. By assumption, $u \in L \backslash M$.

- If $y - x \in M$, $y - \tau_u(x) = y - x \pm u \in L \backslash M$.
- Otherwise, $y - x - (y - x \pm u) = \mp u \in M$.

If at the end of step 2 of the modified sampling procedure there exists a pair $(x, y) \in \mathcal{Z}$ with $x \in C_1 \cup C_2$ and one of the following conditions holds:

- $y - x \in M$ and in step 3 we replace x with $\tau_u(x)$ or
- $y - x \in L \backslash M$ and in step 3 we do not replace x with $\tau_u(x)$,

the modified sampling procedure returns a vector $v \in L \backslash M$. In step 3 of the modified sampling procedure we decide for each pair $(x, y) \in \mathcal{Z}$ uniformly and independently if we replace it or not. Using Lemma 5.1.12 with $\nu = 2^n$, the set $\mathcal{Z}$ contains at least 2^n pairs (x, y) with the property $x \in C_1 \cup C_2$ with probability $1 - 2^{-n+1}$. Therefore, assuming that the set $\mathcal{Z}$ contains at least n such pairs, the probability that the modified sampling procedure does not return a vector $v \in L \backslash M$, is bounded by 2^{-2^n}. Hence, the success probability of the modified sampling procedure is at least $1 - 2^{-\Omega(n)}$. □

The sampling procedure and the modified sampling procedure generate the same output distribution as we have seen in Theorem 5.1.9. Additionally, we have shown in Lemma 5.1.2 that we can restrict ourselves to instances of the generalized shortest vector problem with $2 \leq \lambda_M^{(\|\cdot\|)}(L) < 3$.

Theorem 5.1.14. *There exists a randomized algorithm that for all tractable norms solves* GSVP *with approximation factor $1+\epsilon$ for arbitrary $0 < \epsilon \leq 3/2$ with success probability $1 - 2^{-\Omega(n)}$. The number of arithmetic operations of the algorithm is*

$$\left(\left(2 + \frac{1}{\epsilon}\right)^n \log_2(r)\right)^{\mathcal{O}(1)},$$

where r is an upper bound on the GSVP*-instance. Each number computed by the algorithm has representation size of at most*

$$\left(n \cdot \log_2 \left(2 + \frac{1}{\epsilon}\right) \cdot \log_2(r)\right)^{\mathcal{O}(1)}.$$

Proof. For all tractable norms, we can assume that we are given a GSVP-instance in form of a full-dimensional lattice $L \subseteq \mathbb{Q}^n$ and some subspace $M \subsetneq \operatorname{span}(L)$ satisfying $2 \leq \lambda_M^{(\|\cdot\|)}(L) < 3$, or equivalently

$$\frac{2}{3} < \frac{2}{\lambda_M^{(\|\cdot\|)}(L)} \leq 1,$$

see Lemma 5.1.2. Let $\delta = \epsilon/4$ and define

$$\kappa_0 := \left\lfloor \log_{1+\delta} \frac{2}{3} \right\rfloor \quad \text{and}$$

$$l := \left\lceil \log_{1+\delta} \frac{2}{\lambda_M^{(\|\cdot\|)}(L)} \right\rceil,$$

then $\kappa_0 \leq l \leq 0$ and the parameter $\rho := (1+\delta)^{2-l}$ satisfies Equation (5.9), i.e.,

$$\frac{1}{2}(1+\delta)\lambda_M^{(\|\cdot\|)}(L) \leq \rho \leq \frac{1}{2}(1+\delta)^2\lambda_M^{(\|\cdot\|)}(L).$$

We apply the sampling procedure for each value $\rho = (1+\delta)^{2-l'}$ with $\kappa_0 \leq l' \leq 0$ with the same parameter N as in Theorem 5.1.7 and in Theorem 5.1.13.
Let $v_{l'} \in L \backslash M$ be the lattice vector discovered by the sampling procedure started with $\rho = (1+\delta)^{2-l'}$ if any lattice vector is discovered. The output will be the smallest $v_{l'} \in L \backslash M$. As we have seen, for the unique $l' = l$ such that $\rho = (1+\delta)^{2-l'}$ satisfies the Equation (5.9), the sampling procedure will find a $(1+\epsilon)$-approximation for GSVP with probability $1 - 2^{-\Omega(n)}$, see Theorem 5.1.13.

We apply the sampling procedure roughly

$$\left|\log_{1+\delta}\frac{2}{3}\right|$$

times. By choosing of $\delta = \epsilon/4$ it follows from Theorem 5.1.7 that the number of arithmetic operations is

$$\left|\log_{1+\epsilon}\frac{2}{3}\right| \cdot \left(\log_2\left(2+\frac{1}{\epsilon}\right) \cdot \log_2(r) \cdot n \cdot N\right)^{\mathcal{O}(1)}.$$

By our choice of N, we have

$$\begin{aligned} N &= \left(\frac{1+\delta}{\delta}\right)^n \cdot 2(2^n + z(R_0, \delta)) \\ &= \left(1+\frac{1}{\delta}\right)^n \cdot 2\left(2^n + (\log_2(R_0) + \log_2(1+\frac{2}{\delta}))(2(1+\frac{2}{\delta})+1)^{n+1}\right) \\ &= 2^{\mathcal{O}(n)}\log_2(R_0)\left(1+\frac{2}{\delta}\right)^{\mathcal{O}(n)} \\ &= \log_2(r)^{\mathcal{O}(1)}\left(2+\frac{1}{\epsilon}\right)^{\mathcal{O}(n)}. \end{aligned}$$

Overall, we obtain that the number of arithmetic operations of the sampling procedure is at most

$$\left(\left(2+\frac{1}{\epsilon}\right)^n \log_2(r)\right)^{\mathcal{O}(1)}.$$

The upper bound on the representation size of each number computed by the sampling procedure follows directly from Theorem 5.1.7. □

Combining this result with the reductions presented in Chapter 4 we obtain single exponential time algorithms approximating SVP, SMP, SIVP, and SMP for all tractable norms with approximation factor $1+\epsilon$ for arbitrary $0 < \epsilon < 3/2$.

5.2. Using the sampling procedure for optimal solutions

In this section we show that a variant of the sampling procedure presented before can be used to compute a shortest lattice vector outside a given subspace exactly, provided there do not exist too many short lattice vectors outside the given subspace.

5.2.1. Description and analysis of the sampling procedure for optimal solutions

In the following, we are given a lattice L and some subspace $M \subsetneq \operatorname{span}(L)$. Furthermore, we assume that there exist absolute constants c, ϵ such that the number of lattice vectors

$v \in L\backslash M$ satisfying $\|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ is bounded by 2^{cn}. If so, we are able to show that the sampling procedure presented in the last section satisfies the following property: With probability exponentially close to 1 there exists at least one vector $v \in L\backslash M$ which is represented by 2^n pairs of the set $\mathcal{Z}$ (after the iteration). Using a modified sampling procedure like the one presented in Algorithm 3, we can show that a shortest vector $u \in L\backslash M$ is the difference of such two vectors.

Without loss of generality we can assume that $\epsilon \leq 1/2$. To turn the $(1+\epsilon)$-sampling procedure into an exact algorithm, we use the sampling procedure described in Algorithm 2 with the parameters

$$\begin{aligned} &\delta = \epsilon/4 \text{ and} \\ &N = ((1+\delta)/\delta)^n \, 2 \left(5 \cdot 2^{(c+1)n} + z(R_0, \delta)\right), \end{aligned} \tag{5.12}$$

where $z(R_0, \delta)$ is defined as in (5.6) in Lemma 5.1.6. We only modify the output: We consider the set

$$\mathcal{O} := \{(y_i - x_i) - (y_j - x_j) | (x_i, y_i), (x_j, y_j) \in \mathcal{Z}\}.$$

The output is a shortest lattice vector $v \in \mathcal{O}$ with $v \in L\backslash M$. A complete description of the algorithm is given in Algorithm 4.

The analysis of this sampling procedure and its number of arithmetic operations are the same as in Section 5.1. Obviously, we can modify the sampling procedure in the same way as in Theorem 5.1.9 by using the mapping τ_u with respect to a shortest vector $u \in L\backslash M$. We obtain a modified sampling procedure like the modified sampling procedure described in Algorithm 3 which generates the same output distribution as the original sampling procedure. This modified sampling procedure for optimal solutions is presented in Algorithm 5.

Hence we only need to analyze the success probability of the modified sampling procedure. We show that the modified sampling procedure computes the lattice vector $u \in L\backslash M$ with probability $1 - 2^{-\Omega(n)}$.

In the following, we consider the set $\mathcal{Z}$ after step 2 and before step 3 of the modified sampling procedure. We define the multiset

$$F := \{(x, y) \in \mathcal{Z} | x \in C_1 \cup C_2\} \subseteq \mathcal{Z}. \tag{5.13}$$

If we apply the sampling procedure for optimal solutions with input of a parameter ρ satisfying $(1/2)(1+\delta)\lambda_M^{(\|\cdot\|)}(L) \leq \rho \leq (1/2)(1+\delta)^2\lambda_M^{(\|\cdot\|)}(L)$, each pair $(x, y) \in F$ represents a lattice vector $y - x \in L$ whose length is at most $(1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$, see Theorem 5.1.7. Furthermore we see that the set F contains with probability $1 - 2^{-\Omega(n)}$ at least $5 \cdot 2^{(c+1)n}$ pairs, using Lemma 5.1.12 with $\nu = 5 \cdot 2^{(c+1)n}$. The following lemma shows that at least 2^n of these pairs represent the same lattice vector.

Algorithm 4 The sampling procedure for optimal solutions

```
Input:
    • A lattice basis $B = [b_1, \dots, b_m]$ of a lattice $L$,
    • a subspace $M \subsetneq \mathrm{span}(L)$, and
    • parameters $0 < \delta < 3/2$ and $\rho \geq 1/2$.
Used subroutine: Sieving procedure.
Output: A vector $v \in L \backslash M$ or "failure".
1.  a) Set $R_0 \leftarrow m \cdot \max\{\|b_i\| 1 \leq i \leq m\}$.
    b) Choose $N$ vectors $x_1, \dots, x_N$ uniformly in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$.
    c) Compute $y_i \in \mathcal{P}(B)$ with $y_i \equiv x_i \mod L$ for $i = 1, \dots, N$.
    d) Set $\mathcal{Z} \leftarrow \{(x_1, y_1), \dots, (x_N, y_N)\}$.
    e) Set $R \leftarrow R_0$ and $a \leftarrow 1 + 2/\delta$.
2. While $R > (1 + \delta)\rho$,
    a) apply the sieving procedure to the set $\{y_i | (x_i, y_i) \in \mathcal{Z}\}$ with the parameters
       $a$ and $R$. The result is a set $J$ and a mapping $\sigma$.
    b) Remove all pairs $(x_i, y_i)$ with $i \in J$ from $\mathcal{Z}$ .
    c) Replace each remaining pair $(x_i, y_i) \in \mathcal{Z}$ with $(x_i, y_i - (y_{\sigma(i)} - x_{\sigma(i)}))$.
    d) Set $R \leftarrow R/a + r$.
3. Set $\mathcal{O} := \{(y_i - x_i) - (y_j - x_j) | (x_i, y_i), (x_j, y_j) \in \mathcal{Z}\}$.
   Output a shortest vector $v \in \mathcal{O}$ with $v \notin M$ if such a vector exists. Otherwise,
   the output is "failure".
```

Algorithm 5 The modified sampling procedure for optimal solutions

```
Input:
  • A lattice basis $B = [b_1, \dots, b_m]$ of a lattice $L$,
  • a subspace $M \subsetneq \mathrm{span}(L)$, and
  • parameters $0 < \delta < 3/2$ and $\rho \geq 1/2$.

Used subroutine: Sieving procedure.

Output: A vector $v \in L \backslash M$ or “failure”.

1.  a) Set $R_0 \leftarrow n \cdot \max\{\|b_i\| \mid 1 \leq i \leq n\}$.
    b) Choose $N$ vectors $x_1, \dots, x_N$ uniformly in $\bar{B}_n^{(\|\cdot\|)}(0, \rho)$.
    c) Compute $y_i \in \mathcal{P}(B)$ with $y_i \equiv x_i \mod L$ for $i = 1, \dots, N$.
    d) Set $\mathcal{Z} = \{(x_1, y_1), \dots, (x_N, y_N)\}$.
    e) Set $R \leftarrow R_0$ and $a \leftarrow 1 + 2/\delta$.
2.  While $R > (1 + \delta)\rho$,
    a) apply the sieving procedure to $\{y_i | (x_i, y_i) \in \mathcal{Z}\}$ with the parameters $a$ and
       $R$. The result is a set $J$ and a mapping $\sigma$.
    b) Remove all pairs $(x_i, y_i)$ with $i \in J$ from $\mathcal{Z}$.
    c) For each pair $(x_i, y_i)$, $i \in J$, replace $x_i$ with $\tau_u(x_i)$ with probability $\frac{1}{2}$.
    d) Replace each remaining pair $(x_i, y_i) \in \mathcal{Z}$ with $(x_i, y_i - (y_{\sigma(i)} - x_{\sigma(i)}))$.
    e) Set $R \leftarrow R/a + \rho$.
3.  For each pair $(x_i, y_i) \in \mathcal{Z}$ replace $x_i$ with $\tau_u(x_i)$ with probability $\frac{1}{2}$.
4.  Set $\mathcal{O} := \{(y_i - x_i) - (y_j - x_j) | (x_i, y_i), (x_j, y_j) \in \mathcal{Z}\}$.
    Output a shortest vector $v \in \mathcal{S}$ with $v \notin M$ if such a vector exists. Otherwise,
    the output is “failure”.
```

Lemma 5.2.1. *Let $L \subseteq \mathbb{Q}^n$ be a full-dimensional lattice and $M \subsetneq \mathrm{span}(L)$ be a subspace. Assume that there exist absolute constants c, ϵ such that the number of $v \in L \backslash M$ satisfying $\|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ is bounded by 2^{cn}.*
Consider the modified sampling procedure of optimal solutions, Algorithm 5, with input of the lattice L, the subspace M, and a parameter ρ satisfying

$$\frac{1}{2} \leq \rho \leq \frac{1}{2} \cdot (1+\delta)^2 \lambda_M^{(\|\cdot\|)}(L),$$

where $0 < \delta \leq \epsilon/4$. Assume that the multiset F defined as in (5.13) contains at least $5 \cdot 2^{(c+1)n}$ pairs. For $v \in L$ we set

$$F_v := \{(x_i, y_i) \in F | y_i - x_i = v\}.$$

Then, there exists a vector $v \in L$ with $|F_v| \geq 2^n$.

Proof. Assuming that $|F_v| < 2^n$ for all lattice vectors $v \in L$, we will derive a contradiction.
In the following, we consider the set of all lattice vectors in L which are represented by a pair $(x, y) \in F$,

$$G := \{v \in L | \exists (x, y) \in F \text{ with } v = y - x\}.$$

Since we assume that $|F| > 5 \cdot 2^{(c+1)n}$ and $|F_v| < 2^n$ for all $v \in L$, we obtain

$$|G| \geq 5 \cdot 2^{c \cdot n}.$$

Since the parameter ρ satisfies $1/2 \leq \rho \leq (1/2) \cdot (1+\delta)^2 \lambda_M^{(\|\cdot\|)}(L)$, it is guaranteed by Theorem 5.1.7 that all lattice vectors in G have length of at most $(1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$. Hence, if we define

$$G_M := G \cap M,$$

the set $G \backslash G_M$ consists of lattice vectors in $L \backslash M$ of length of at most $(1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$. By assumption, $|G \backslash G_M| \leq 2^{cn}$ and therefore

$$|G_M| = |G| - |G \backslash G_M| \geq 5 \cdot 2^{c \cdot n} - 2^{c \cdot n} = 2^{c \cdot n + 2}. \tag{5.14}$$

Every vector $v \in G_M$ is represented by a pair (x, y) with $x \in C_1 \cup C_2$ and $v = y - x$. Since $\tau_u(x) \in \bar{B}_n^{(\|\cdot\|)}(0, \rho)$, we can see that $\|y - \tau_u(x)\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$, analogously to the proof of Theorem 5.1.7. Furthermore, we have

$$y - \tau_u(x) = y - x \pm u = v \pm u \in L \backslash M.$$

Since τ_u is injective, this shows that we can define an injective mapping

$$G_M \to \{v \in L \backslash M | \|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)\},\ (x, y) \mapsto y - \tau_u(x)$$

and it follows that

$$|G_M| \leq |\{v \in L\backslash M \mid \|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)\}|.$$

Combining this with (5.14), we obtain that the number of lattice vectors $v \in L\backslash M$ satisfying $\|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ is at least $2^{c\cdot n+2}$, i.e.,

$$|\{x \in L\backslash M|(1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)\}| \geq 2^{c\cdot n+2}.$$

This contradicts the assumption that the number of lattice vectors $v \in L\backslash M$ satisfying $\|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ is bounded by $2^{c\cdot n}$. □

Using this lemma, we get:

Theorem 5.2.2. *Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{Q}^n$ be a full-dimensional lattice and $M \subsetneq \operatorname{span}(L)$ be a subspace. Assume that there exist absolute constants c, ϵ such that the number of $v \in L\backslash M$ satisfying $\|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ is bounded by 2^{cn}. Then, the modified sampling procedure for optimal solutions, Algorithm 5, solves the generalized shortest vector problem with success probability $1 - 2^{-\Omega(n)}$.*

Proof. Using Lemma 5.1.12 with $\nu = 5\cdot 2^{(c+1)n}$, we obtain that with probability $1-2^{-\Omega(n)}$ the set F defined as in (5.13) contains at least $5\cdot 2^{(c+1)n}$ pairs. In this case there exists a lattice vector $v \in L$ with $|F_v| \geq 2^n$, see Lemma 5.2.1. In step 3 of the modified sampling procedure we decide for each pair $(x,y) \in F_v$ uniformly at random whether we replace x with $\tau_u(x)$ or not. If there exist $(x_i,y_i),(x_j,y_j) \in F_v$ such that in step 3 the mapping τ is applied to x_i but not to x_j then $u \in \mathcal{O}$. Since we decide uniformly whether we replace x with $\tau_u(x)$ this event happens with probability at least $1 - 2\cdot 2^{-2^n}$. □

Like in Theorem 5.1.9, we can show that the sampling procedure for optimal solutions, Algorithm 4, and the modified sampling procedure for optimal solutions, Algorithm 5, generate the same output distribution.

Theorem 5.2.3. *(Theorem 5.0.3 restated.)*
Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$. Let $L \subseteq \mathbb{Q}^n$ be a full-dimensional lattice and $M \subsetneq \operatorname{span}(L)$ be a subspace. Assume that there exist absolute constants c, ϵ such that the number of $v \in L\backslash M$ satisfying $\|v\| \leq (1+\epsilon)\lambda_M^{(\|\cdot\|)}(L)$ is bounded by 2^{cn}. Then, there exists an algorithm that solves the generalized shortest vector problem with success probability $1 - 2^{-\Omega(n)}$. The number of arithmetic operations of the algorithm is $(2^n \cdot \log_2(r))^{\mathcal{O}(1)}$, where r is an upper bound on the size of the lattice and the subspace. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most $(n\cdot\log_2(r))^{\mathcal{O}(1)}$.

5.2.2. Consequences for other lattice problems

In this section, we show that we can use this result to obtain probabilistic single exponential time algorithms for Svp, Smp, Sivp, and Cvp. For this, we need to show for each problem that the number of $(1+\epsilon)$-approximate solutions is single exponential.

To obtain an upper bound for the number of $(1+\epsilon)$-approximate solutions for these lattice problems we use our results from Chapter 4, where we have seen that the number of lattice vectors in a ball with radius R is essentially single exponential in the relation between the radius R and the minimum distance of the lattice, see Lemma 4.2.11.

For the shortest vector problem we can use this result to show that the assumptions of Theorem 5.0.3 (Theorem 5.2.3 respectively) are always satisfied. For the successive minima problem we can use the same result, but here the assumptions of Theorem 5.0.3 are only satisfied in the special cases where the relation between the n-th successive minimum and the minimum distance of the lattice is not too large. For the closest vector problem, we have to go back to the original reduction presented in Section 4.3.1 to see that in some special cases the assumptions of Theorem 5.0.3 are satisfied.

Theorem 5.2.4. *(Theorem 5.0.4 restated.)*
For all tractable norms, there exists a randomized algorithm that solves the shortest vector problem with success probability $1-2^{-\Omega(n)}$. The number of arithmetic operations of the algorithm is $(2^n\log_2(r))^{\mathcal{O}(1)}$, where n is the rank of the lattice and r is an upper bound on its size. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most $(n\cdot\log_2(r))^{\mathcal{O}(1)}$.

Proof. Given a lattice L, we set $M:=\{0\}$. Then, the subspace avoiding minimum corresponds to the minimum distance of the lattice, i.e., $\lambda_M^{(\|\cdot\|)}(L)=\lambda_1^{(\|\cdot\|)}(L)$. We have seen in Chapter 4 that the number of lattice vectors in L with length at most $(1+\epsilon)\lambda_1^{(\|\cdot\|)}(L)$ is upper bounded by

$$\left|\bar{B}_n^{(\|\cdot\|)}\left(0,(1+\epsilon)\lambda_1^{(\|\cdot\|)}(L)\right)\cap L\right|\leq(2(1+\epsilon)+1)^n=(3+2\epsilon)^n=2^{cn}$$

for a $c\in\mathbb{N}$, see Corollary 4.2.12. With Theorem 5.0.3, we obtain that the sampling procedure for optimal solutions with input of the lattice L, the subspace M, and the parameter N chosen as in (5.12) computes a vector $v\in L\backslash M$ with $\|v\|\leq\lambda_M^{(\|\cdot\|)}(L)$ with probability exponentially close to 1 and therefore a shortest non-zero lattice vector in L. □

Similarly, the sampling method for optimal solutions can be used to compute the successive minima of a lattice L exactly provided that the n-th successive minimum $\lambda_n^{(\|\cdot\|)}(L)$ is bounded by $c\lambda_1^{(\|\cdot\|)}(L)$ for some constant c. The proof of the following result is based on the same idea as the reduction from SMP to GSVP, see Theorem 4.3.3.

Theorem 5.2.5. *(Theorem 5.0.5 restated.)*
Let $\|\cdot\|$ be a tractable norm on $\mathbb{R}^n$ and $L\subseteq\mathbb{Q}^n$ be a full-dimensional lattice. Assume that the n-th successive minimum $\lambda_n^{(\|\cdot\|)}(L)$ is bounded by $c\cdot\lambda_1^{(\|\cdot\|)}(L)$ for some constant $c\in\mathbb{N}$. Then, with success probability $1-2^{-\Omega(n)}$, the successive minima of L can be computed using $(2^n\cdot\log_2(r))^{\mathcal{O}(1)}$ arithmetic operations, where r is an upper bound on

the size of the lattice. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most $(n \cdot \log_2(r))^{\mathcal{O}(1)}$.

Proof. We have seen in Theorem 5.2.4 that we are able to use the sampling procedure for optimal solutions to get a shortest non-zero vector in L. Given $v_1, \ldots, v_{i-1} \in L$ linearly independent for $i > 1$, we consider the subspace $M := \text{span}(v_1, \ldots, v_{i-1})$. Then we get $\lambda_M^{(\|\cdot\|)}(L) \leq \lambda_i^{(\|\cdot\|)}(L)$. Since $\lambda_n^{(\|\cdot\|)}(L) \leq c \cdot \lambda_1^{(\|\cdot\|)}(L)$, for all $\epsilon > 0$, the number of lattice vectors in L of length of at most $(1+\epsilon)\lambda_i^{(\|\cdot\|)}(L)$ is upper bounded by

$$\left|\bar{B}_n^{(\|\cdot\|)}\left(0, (1+\epsilon)\lambda_i^{(\|\cdot\|)}(L)\right) \cap L\right| \leq (2(1+\epsilon)c + 1)^n,$$

see Corollary 4.2.13. With Theorem 5.0.3 we get that the sampling procedure for optimal solutions with input of the lattice L, the subspace M and the parameter N defined as in (5.12) computes a vector $v_i \in L \backslash M$ with $\|v_i\| \leq \lambda_M^{(\|\cdot\|)}(L)$. □

For the closest vector problem, we obtain a similar result for instances of CVP where the distance between the target vector and the lattice is at most c times the minimum distance of the lattice for some fixed constant $c > 0$.

Theorem 5.2.6. *(Theorem 5.0.6 restated.)*
Let $\|\cdot\|$ *be a tractable norm on* $\mathbb{R}^n$. *Let* $L \subseteq \mathbb{Q}^n$ *be a full-dimensional lattice and* $t \in \text{span}(L) \cap \mathbb{Q}^n$ *be some target vector. Assume that there exists some absolute constant* c *such that* $\mu^{(\|\cdot\|)}(t, L) \leq c \cdot \lambda_1^{(\|\cdot\|)}(L)$. *Then, a vector* $v \in L$ *satisfying* $\|t - v\| = \mu^{(\|\cdot\|)}(t, L)$ *can be computed using* $(2^n \cdot \log_2(r))^{\mathcal{O}(1)}$ *arithmetic operations, where* r *is an upper bound on the size of the* CVP*-instance. The algorithm runs in single exponential space and each number computed by the algorithm has representation size of at most* $(n \cdot \log_2(r))^{\mathcal{O}(1)}$.

To prove this theorem, we follow the reduction from CVP to GSVP presented in Section 4.3.1 in Chapter 4. Given a lattice $L \subseteq \mathbb{R}^n$ and a target vector $t \in \mathbb{R}^n$ together with a parameter $\alpha > 0$, we consider the unique parameter ρ such that

$$\rho \leq \mu^{(\|\cdot\|)}(t, L) \leq (1+\alpha)\rho.$$

We use the same lifting technique and the same parameter

$$\gamma := \frac{1+\epsilon}{1-\epsilon}(1+\alpha)\rho \tag{5.15}$$

with $0 < \epsilon < 1/2$ to define the $(n+1)$-dimensional lattice

$$L' := \mathcal{L}\left(\begin{pmatrix} b_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} b_m \\ 0 \end{pmatrix}, \begin{pmatrix} t \\ \gamma \end{pmatrix}\right) \tag{5.16}$$

and the subspace

$$M := \text{span}\left(\begin{pmatrix} b_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} b_m \\ 0 \end{pmatrix}\right) \subsetneq \text{span}(L'), \tag{5.17}$$

where $B = [b_1, \ldots, b_m] \in \mathbb{R}^{n \times m}$ is a basis of the lattice L. Also, we consider the same norm F on $\mathbb{R}^{n+1}$ which is defined as $F(x) = \|\bar{x}\| + |\hat{x}|$ for $x = (\bar{x}^T, x^T) \in \mathbb{R}^{n+1}$. As we have seen in Lemma 4.3.4, the norm F is tractable, since $\|\cdot\|$ is a tractable norm.

Now, the main part of the proof of Theorem 5.2.6 is to show that it follows from $\mu^{(\|\cdot\|)}(t, L) \leq c \cdot \lambda_1^{(\|\cdot\|)}(L)$ that there do not exist too many lattice vectors in $L' \backslash M$ whose length with respect to the norm F is at most $(1+\epsilon)\lambda_M^{(F)}(L')$, i.e., to show that there exists a constant $c_1 \in \mathbb{R}$ such that

$$\left|\bar{B}_n^{(F)}(0, (1+\epsilon)\lambda_M^{(F)}(L')) \cap (L' \backslash M)\right| \leq 2^{c_1 n}.$$

If we can show this, we can use the sampling procedure for optimal solutions to obtain a shortest vector $v \in L' \backslash M$ with respect to the norm F with success probability $1 - 2^{-\Omega(n)}$. As we have seen in the reduction from CVP to GSVP, such a vector is of the form $v = \pm(z - t, -\gamma)$, where $z \in L$ is a lattice vector in L that is closest to the target vector t with respect to the norm $\|\cdot\|$, see Lemma 4.3.8 in Chapter 4.

We consider a parameter ρ satisfying $\rho \leq \mu^{(\|\cdot\|)}(t, L)$. We have already seen that under this assumption the subspace avoiding minimum $\lambda_M^{(F)}(L')$ of the lattice L' and the subspace $M \subsetneq \text{span}(L')$ in the norm F is less than 2γ,

$$\lambda_M^{(F)}(L') < \frac{2}{1+\epsilon}\gamma < 2\gamma, \tag{5.18}$$

see Claim 4.3.7 in Chapter 4. Hence, it follows from the definition of the norm F that the minimum distance of the lattice L' with respect to the norm F is the minimum of the minimum distance of the lattice L with respect to the norm $\|\cdot\|$ and the value $\mu^{(\|\cdot\|)}(t, L) + \gamma$, i.e.,

$$\lambda_1^{(F)}(L') = \min\left\{\lambda_1^{(\|\cdot\|)}(L), \mu^{(\|\cdot\|)}(t, L) + \gamma\right\}.$$

By assumption, there exists some absolute constant c such that $\mu^{(\|\cdot\|)}(t, L) \leq c \cdot \lambda_1^{(\|\cdot\|)}(L)$. If $\lambda_1^{(F)}(L') = \lambda_1^{(\|\cdot\|)}(L)$, we have $\mu^{(\|\cdot\|)}(t, L) \leq c \cdot \lambda_1^{(\|\cdot\|)}(L) = c \cdot \lambda_1^{(F)}(L')$. Otherwise, we obtain $\mu^{(\|\cdot\|)}(t, L) \leq \mu^{(\|\cdot\|)}(t, L) + \gamma = \lambda_1^{(F)}(L')$. This shows that in both cases the parameter ρ satisfies

$$\rho \leq \mu^{(\|\cdot\|)}(t, L) \leq c \cdot \lambda_1^{(F)}(L').$$

In the next lemma, we will see that this guarantees that there do not exist too many lattice vectors in L' with length 2γ in the norm F.

Lemma 5.2.7. *Let $L \subseteq \mathbb{R}^n$ be a lattice and $t \in \text{span}(L)$ be some target vector. For $\alpha > 0$ and $0 < \epsilon < 3/2$ define the parameter γ and the lattice $L' \subseteq \mathbb{R}^{n+1}$ as above, see (5.15) and (5.16).*

Assume furthermore that there exists a parameter $\rho > 0$ and some absolute constant c such that

$$\rho < c \cdot \lambda_1^{(F)}(L').$$

Then, the number of lattice vectors in L' with length of at most 2γ is upper bounded by

$$\left|\bar{B}_n^{(F)}(0, 2\gamma) \cap L'\right| \leq 2^{c_1 n}$$

for some constant $c_1 \in \mathbb{N}$.

Proof. Using Lemma 4.2.11 with the radius $R = 2\gamma$, we obtain that the number of lattice vectors in L' whose length with respect to the norm F is at most 2γ is upper bounded by

$$\left|\bar{B}_n^{(F)}(0, 2\gamma) \cap L'\right| \leq \left(\frac{4\gamma + \lambda_1^{(F)}(L')}{\lambda_1^{(F)}(L')}\right)^n.$$

By definition of γ and using that $\rho \leq c \cdot \lambda_1^{(F)}(L')$, we obtain

$$\begin{aligned}\left|\bar{B}_n^{(F)}(0, 2\gamma) \cap L'\right| &\leq \left(\frac{4\frac{1+\epsilon}{1-\epsilon}(1+\alpha)\rho + \lambda_1^{(F)}(L')}{\lambda_1^{(F)}(L')}\right)^n \\ &\leq \left(\frac{\left(4\frac{1+\epsilon}{1-\epsilon}(1+\alpha)c + 1\right)\lambda_1^{(F)}(L')}{\lambda_1^{(F)}(L')}\right)^n \\ &\leq 2^{c_1 n}\end{aligned}$$

for some constant $c_1 \in \mathbb{N}$. □

As we have seen in (5.18), the subspace avoiding minimum of the lattice L' and the subspace M is less than $\lambda_M^{(F)}(L') < (2/(1+\epsilon))\gamma$. Thus, it follows from Lemma 5.2.7 that

$$\left|\bar{B}_n^{(F)}(0, (1+\epsilon)\lambda_M^{(F)}(L')) \cap L'\right| \leq \left|\bar{B}_n^{(F)}(0, 2\gamma) \cap L'\right| \leq 2^{c_1 n}.$$

In particular, the number of vectors $v \in L' \backslash M$ satisfying $F(v) \leq (1+\epsilon)\lambda_M^{(F)}(L')$ is bounded by $2^{c_1 n}$ for some fixed constant $c_1 \in \mathbb{N}$. Hence, the assumptions of Theorem 5.0.3 are satisfied, that means the lattice membership algorithm for optimal solutions with input of the lattice L' and the subspace M defined as in (5.16) and (5.17) using a parameter ρ satisfying $\rho < \mu^{(\|\cdot\|)}(t, L) < (1+\alpha)\rho$ computes a shortest lattice vector $u \in L' \backslash M$ with respect to the norm F with probability exponentially close to 1. As we have seen in Lemma 4.3.8, the vector u is of the form $u = \pm(z - t, -\gamma)$, where $z \in L$ is a lattice vector that is closest to the target vector t with respect to the norm $\|\cdot\|$. This proves Theorem 5.2.6.

5.3. Discussion of the results

In this chapter, we presented a probabilistic single exponential time algorithm based on the AKS-sampling technique that approximates the generalized shortest vector problem with approximation factor $1+\epsilon$ for arbitrary $0 < \epsilon < 3/2$. Unfortunately, we are not able to use this technique to solve the generalized shortest vector problem exactly. Moreover, it seems to be impossible to develop an algorithm based on the AKS-sampling technique that solves one of the lattice problems SMP, SIVP, or CVP exactly. If we want to use the sampling technique to obtain exact solutions we need to find some kind of collisions. That means, the number N of vectors which are chosen in the beginning of the sampling procedure needs to be large enough to guarantee that at the end of the algorithm we have two vectors of the form x and $x+u$, where u is an optimal solution of the corresponding lattice problem and both x and $x+u$ are guaranteed to be approximate solutions. But we have already seen in Chapter 4 that for the lattice problems SMP, SIVP, and CVP, the number of approximate solutions can be arbitrarily large.

In 2008, Arvind and Joglekar improved our approximation algorithm for GSVP with respect to the Euclidean norm such that the number of arithmetic operations of the algorithm is $(2^n(1/\epsilon)^k \log_2(r))^{\mathcal{O}(1)}$, where n is the rank of the lattice, k is the dimension of the subspace, and r is an upper bound on the size of the lattice and the subspace, see [AJ08]. The improvement of the running time is achieved as follows: In their algorithm, the number N of vectors chosen at the beginning of the sampling procedure needs to be large enough such that after the iterations there exist enough pairs $(x_i, y_i), (x_j, y_j)$ with $x_i, x_j \in C_1 \cup C_2$ satisfying $\|(x_i - y_i) - (x_j - y_j)\| \leq \epsilon$ and $(x_i - y_i) - (x_j - y_j) \in M$. To determine a corresponding parameter N they use a packing argument in $\mathbb{R}^k$, where k is the dimension of the subspace M. Thus, they use a bijective linear transformation between the subspace M and the vector space $\mathbb{R}^k$. Since this bijective linear transformation is length preserving with respect to the Euclidean norm they can use an argument like Lemma 4.2.11 in Chapter 4 to obtain an upper bound on the number of different lattice vectors in $L \cap M$ of Euclidean length at most ϵ. This upper bound depends only on the dimension k of the subspace and not on the dimension of span(L). Unfortunately, we do not know how to construct a bijective linear transformation from an arbitrary k-dimensional subspace to the vector space $\mathbb{R}^k$ which is length preserving with respect to some arbitrary but fixed norm.

6. A deterministic algorithm for the lattice membership problem

In this chapter, we consider algorithmic solutions for the lattice membership problem. To recall, in the lattice membership problem we are given a full-dimensional bounded convex set together with a lattice. The goal is to compute a lattice vector in the convex set or to decide that the convex set does not contain a lattice vector, see Definition 4.3.12 in Chapter 4.

We show that there exists a deterministic algorithm that solves the lattice membership problem in polynomial space for all ℓ_p-balls and polytopes. If we consider ℓ_p-norms, $1 < p < \infty$, we obtain an algorithm whose number of arithmetic operations is $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r is an upper bound on the size of the coefficients defining the convex set and n is the dimension of the ℓ_p-ball. For all polyhedral norms, we obtain an algorithm whose number of arithmetic operations is $(s \cdot \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r and n are defined as above and s is the number of constraints defining the polytope. In particular, for the ℓ_1-norm and the ℓ_∞-norm, we obtain an algorithm whose number of arithmetic operations is $\log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$.

In Chapter 4, we have seen that the lattice membership problem can be seen as a geometric reformulation of the closest vector problem, i.e., that there exists a polynomial time reduction from the closest vector problem to the lattice membership problem for all ℓ_p-norms and all polyhedral norms. This leads to a deterministic polynomially space bounded algorithm for the closest vector problem for all ℓ_p-norms and all polyhedral norms.

As we have seen in Chapter 4, the lattice membership problem is a generalization of the integer programming feasibility problem from polytopes to general bounded convex sets. Hence, the existence of algorithmic solutions for the lattice membership problem is closely related to the existence of algorithmic solutions for the integer programming feasibility problem. Our algorithm is a variant of Lenstra's algorithm for integer programming used together with a variant of the ellipsoid method, see [Len83]. To guarantee that the algorithm runs in polynomial space, we use a preprocessing method from Frank and Tardos developed for Lenstra's algorithm for integer programming, see [FT87].

To put our results in perspective, we shortly review the major results based on Lenstra's technique in the following.

Lenstra's algorithm for integer programming and related results

In 1979, Lenstra presented the first polynomial time algorithm that solves the integer programming feasibility problem in fixed dimension, [Len83]. This algorithm was improved by Kannan in 1987, [Kan87b]. Considering the dimension as a part of the input, the number of arithmetic operations of this algorithm is $\mathcal{O}(n^{(5/2)n} \log_2(r))$, where r is an upper bound on the size of the input polytope. Hence, our result improves the running time of Lenstra's algorithm by the factor $n^{n/2}$ while keeping polynomial space complexity.

In 2005, Heinz generalized Lenstra's algorithm to obtain an algorithm for integer optimization over quasiconvex polynomials, [Hei05]. To recall, a function $f : \mathbb{R}^n \to \mathbb{R}$ is quasiconvex if all α-sublevel sets $\{x \in \mathbb{R}^n | f(x) \leq \alpha\}$, $\alpha \in \mathbb{R}$, are convex sets, see Definition 2.1.7 in Chapter 2. Heinz considered quasiconvex polynomials $F_1, \dots, F_m \in \mathbb{Z}[x]$, which define a convex set $Y := \{x \in \mathbb{R}^n | F_i(x) < 0 \text{ for all } 1 \leq i \leq m\}$. His algorithm either computes an integer vector in this set or shows that this set does not contain an integer vector. Recently, this algorithm was improved by Hildebrand and Köppe, who presented an algorithm for this problem using $m \cdot \log_2(r)^{\mathcal{O}(1)} d^{\mathcal{O}(n)} n^{(2+o(1))n}$ arithmetic operations, where d is an upper bound on the total degree of the m polynomials and r is an upper bound on the size of the input, see [HK10].
In particular, their algorithm can be used to decide whether the set $\{x \in \mathbb{R}^n | \|x-t\|_p^p - \alpha < 0\}$ with $t \in \mathbb{R}^n$ and $\alpha > 0$ contains a lattice vector if p is an even number, since for even p the function $\| \cdot \|_p^p : \mathbb{R}^n \to \mathbb{R}$, $x \mapsto \|x\|_p^p = \sum_{i=1}^n x_i^p$ is a quasiconvex polynomial and obviously for a given vector $t \in \mathbb{R}^n$ and radius $\alpha > 0$, we have

$$B_n^{(p)}(t, \alpha) = \{x \in \mathbb{R}^n | \|x - t\|_p < \alpha\} = \{x \in \mathbb{R}^n | \|x - t\|_p^p - \alpha^p < 0\}.$$

If p is not an even number, the function $\| \cdot \|_p^p : \mathbb{R}^n \to \mathbb{R}$, $x \mapsto \|x\|_p^p$ is not even a polynomial. Although it is possible to represent $B_n^{(p)}(t, \alpha)$ as the intersection of polynomials, these polynomials are not quasiconvex. For example, $B_n^{(3)}(0, 1)$ can be represented using the sublevel sets $\{x \in \mathbb{R}^2 | x_1^3 + x_2^3 - 1 < 0\}$, $\{x \in \mathbb{R}^2 | -x_1^3 + x_2^3 - 1 < 0\}$, $\{x \in \mathbb{R}^2 | -x_1^3 - x_2^3 - 1 < 0\}$, and $\{x \in \mathbb{R}^2 | x_1^3 - x_2^3 - 1 < 0\}$, but we have already seen that for example the function $x \mapsto x_1^3 + x_2^3 - 1$ is not quasiconvex, see Figure 2.3 on page 14 in Chapter 2. Thus, the result of Heinz cannot be applied directly to achieve our results. Additionally, their algorithm has the disadvantage of not being polynomially space bounded.

In 2010, Dadush, Peikert, and Vempala presented a randomized algorithm for the lattice membership problem for well-bounded convex bodies given by a separation oracle, see [DPV11] and [DPV10]. Their algorithm is also based on Lenstra's algorithm for integer programming. The expected number of arithmetic operations of this algorithm is $\mathcal{O}(n^{(4/3)n}) \log_2(r)^{\mathcal{O}(1)}$. Their algorithm uses so-called M-ellipsoids. In [DPV11] they present an algorithm that computes an M-ellipsoid for a well-bounded convex body in expected single exponential time. Recently, Dadush and Vempala described a deterministic algorithm that computes an approximate M-ellipsoid for any well-bounded convex body in time $\mathcal{O}(\log_2(n))^n$, see [DV12]. This yields a deterministic algorithm for the lattice membership problem for well-bounded convex bodies given by a separation oracle where the number of arithmetic operations is $\mathcal{O}(n^{(4/3)n}) \log_2(r)^{\mathcal{O}(1)}$. Of course, the number of arithmetic operations of their algorithm is better than ours, but compared to our result, their algorithm has the disadvantage of having exponential space complexity.

Main results of this chapter and its consequences for other lattice problems

In this chapter, we present a deterministic algorithm that solves the lattice membership problem in polynomial space for all ℓ_p-balls, $1 < p < \infty$, and polytopes.

Theorem 6.0.1. *There exists a deterministic algorithm that solves the lattice membership problem for all convex sets generated by an ℓ_p-norm, $1 < p < \infty$, or a polyhedral norm.*

- *If the convex set is an ℓ_p-ball with $1 < p < \infty$, the number of arithmetic operations of the algorithm is at most $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$. The algorithm runs in polynomial space and each number computed by the algorithm has bit size of at most $p \cdot n^{\mathcal{O}(1)} \log_2(r)$.*
- *If the convex set is a full-dimensional polytope symmetric about the origin given by s constraints, the number of arithmetic operations is at most $(s \cdot \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$. The algorithm runs in polynomial space and each number computed by the algorithm has bit size of at most $n^{\mathcal{O}(1)} \log_2(r)$.*

We present the algorithm as a general algorithmic framework. This framework works for all full-dimensional bounded convex sets which are contained in some class $\mathcal{K}$ such that there exists a so-called flatness algorithm for this class. Loosely speaking, a flatness algorithm for such a class of bounded convex sets computes a bounded number of parallel affine hyperplanes for a given convex set from this class such that the following holds: The convex set contains a lattice vector if and only if the intersection of the convex set with one of these affine hyperplanes contains a lattice vector. That means, the flatness algorithm reduces the solution of the lattice membership problem for a bounded convex set of dimension n to several solutions of the lattice membership problem of convex sets of dimension $n - 1$.

To obtain an algorithm that solves the lattice membership problem for polytopes, we consider the class of full-dimensional polytopes and show that for this class there exists a flatness algorithm. If we want to obtain an algorithm that solves the lattice membership problem for ℓ_p-balls with $1 < p < \infty$ there arises the technical difficulty that we are not able to develop a flatness algorithm for ℓ_p-balls since the class of ℓ_p-balls is not closed under bijective affine transformation and intersection with hyperplanes. Due to this reason, we consider a generalization of ℓ_p-balls, the class of so-called ℓ_p-bodies. For this class, we show that there exists a flatness algorithm. This part is the main technical contribution of our lattice membership algorithm.

In Section 4.3 of Chapter 4, we have seen that the lattice membership problem is a geometric reformulation of the closest vector problem. Particularly, we have seen that there exists a polynomial time reduction from the closest vector problem to the lattice membership problem for all ℓ_p-norms and all polyhedral norms, see Proposition 4.3.13 in Chapter 4. Combining this with Theorem 6.0.1, it implies a deterministic polynomially space bounded algorithm that solves the closest vector problem with respect to an ℓ_p-norm, $1 < p < \infty$, and a polyhedral norm, e.g. the ℓ_1-norm and the ℓ_∞-norm.

Theorem 6.0.2. *There exists a deterministic polynomially space bounded algorithm that solves the closest vector problem for all ℓ_p-norms, $1 < p < \infty$, and all polyhedral norms, e.g. the ℓ_1-norm and the ℓ_∞-norm.*

- *For all ℓ_p-norms with $1 < p < \infty$, the number of arithmetic operations of the algorithm is $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$.*

- *For all polyhedral norms, the number of arithmetic operations of the algorithm is $(s \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$, where s is the number of constraints defining the polytope. In particular, for the ℓ_1-norm and the ℓ_∞-norm we obtain an algorithm for* CVP, *where the number of arithmetic operations is $\log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$.*

Here, r is an upper bound on the size of the CVP*-instance and n is the dimension of the vector space.*

Organization This chapter is organized as follows: We start with a description of Lenstra's algorithm as a general framework for algorithmic solutions of the lattice membership problem. This is done in Section 6.1. Here, we consider some unspecified class $\mathcal{K}$ of full-dimensional bounded convex sets, for which we assume the existence of a flatness algorithm. Then, we adapt this framework to concrete classes of full-dimensional bounded convex sets. In Section 6.2, we consider full-dimensional polytopes and in Section 6.3 generalizations of ℓ_p-balls, where $1 < p < \infty$.
To complete the description of the lattice membership algorithm, we describe in Section 6.4 how a flatness algorithm for these classes of convex sets can be realized. We start with a description of a flatness algorithm for ellipsoids. Then we generalize this result to general bounded convex sets, for which we are able to compute an approximate Löwner-John ellipsoid. For polytopes and the generalization of ℓ_p-balls, we are able to compute an approximate Löwner-John ellipsoid and we obtain a flatness algorithm for these classes of convex sets. At the end of this chapter, we describe a slight modification of the replacement procedure due to Frank and Tardos, which can be used to guarantee that our lattice membership algorithm runs in polynomial space.

6.1. A general algorithm for the lattice membership problem

In this section, we describe a general framework to solve the lattice membership problem for full-dimensional bounded convex sets and full-dimensional lattices. Before we present a concrete and detailed description of the lattice membership algorithm, we start with the description of the main geometric idea behind the algorithm.

The lattice membership algorithm is a recursive algorithm which works for classes of bounded convex sets. Since we describe a general framework here, we do not specify how the convex sets from the class $\mathcal{K}$ are given.

The class $\mathcal{K}$ need to satisfy certain properties. We will define and explain these properties at a suitable place where they are necessary for the development of the membership algorithm. First of all, we assume only that the class $\mathcal{K}$ is closed under bijective linear transformation. Then it is enough to solve the lattice membership problem for those instances where the corresponding lattice is the integer lattice $\mathbb{Z}^n$. Since every vector from a lattice $L = \mathcal{L}(B)$ is an integer linear combination of the basis vectors of B, any bounded convex set $\mathcal{C} \subseteq \text{span}(L)$ contains a lattice vector from L if and only if the bounded convex set $B^{-1}\mathcal{C}$ contains an integer vector.

Lemma 6.1.1. *Let $L \subseteq \mathbb{R}^n$ be a full-dimensional lattice given by a basis $B \in \mathbb{R}^{n\times n}$ and $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set. Then, the convex set $\mathcal{C}$ contains a lattice vector from L if and only if the convex set $B^{-1} \cdot \mathcal{C} = \{B^{-1}x | x \in \mathcal{C}\}$ contains an integer vector.*

6.1.1. The main idea of the lattice membership algorithm

The lattice membership algorithm uses the concept of branch and bound. Given a bounded convex set $\mathcal{C}$ from the class $\mathcal{K}$ we consider a family of affine hyperplanes given by a vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$, $\bigcup_{k\in\mathbb{Z}} H_{k,\tilde{d}}$. Obviously, every integer vector $v \in \mathbb{Z}^n$, which is contained in $\mathcal{C}$, satisfies $\langle \tilde{d}, v\rangle = k$ for some integer value $k \in \mathbb{Z}$ and k is contained in the interval

$$I_{\mathcal{C}} := \left[\inf\{\langle \tilde{d}, x\rangle | x \in \mathcal{C}\}, \sup\{\langle \tilde{d}, x\rangle | x \in \mathcal{C}\}\right].$$

Hence, to decide whether the bounded convex set $\mathcal{C}$ contains an integer vector, it is sufficient to consider all integer values k which are contained in the interval $I_{\mathcal{C}}$ and check recursively whether the convex set $\mathcal{C} \cap H_{k,\tilde{d}}$ contains an integer vector. This idea is illustrated in Figure 6.1. In the following, we will call an algorithm which realizes this idea a lattice membership algorithm.

Since the convex set $\mathcal{C}$ can be arbitrarily large, we cannot generally assume that the length of the interval $I_{\mathcal{C}}$ is bounded. But we will show that we can restrict ourselves to consider only a bounded number of affine hyperplanes. That means we can show that there exists a non-decreasing function $f : \mathbb{N} \to \mathbb{R}$ such that for every bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n \cap \mathcal{K}$ of dimension m there exists a vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ and an interval $I_{\mathcal{C}}$ of length at most $f(m)$ such that the following holds: The convex set $\mathcal{C}$ contains an integer vector if and only if there exists an integer $k \in \mathbb{Z} \cap I_{\mathcal{C}}$ such that the convex set $\mathcal{C} \cap H_{k,\tilde{d}}$ contains an integer vector.
We call a vector $\tilde{d}$ that satisfies this property a $f(m)$-flatness direction of the convex set $\mathcal{C}$.

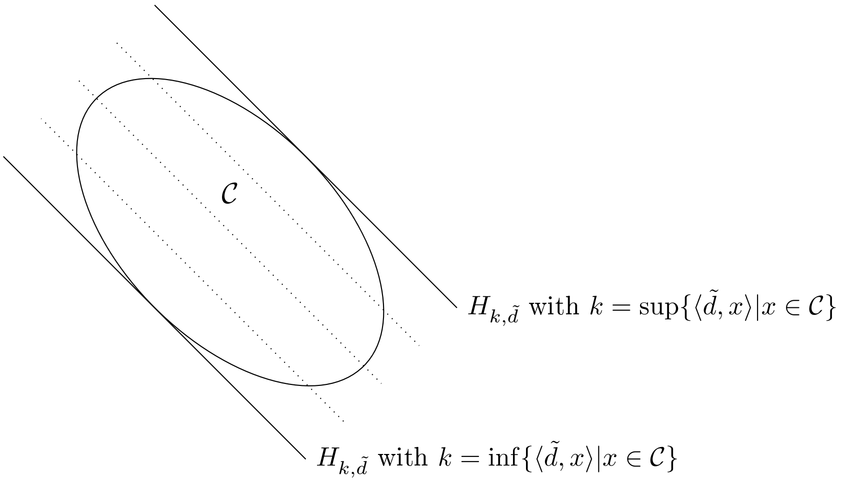
$\mathcal{C}$
$H_{k,\tilde{d}}$ with $k = \sup\{\langle \tilde{d}, x\rangle | x \in \mathcal{C}\}$
$H_{k,\tilde{d}}$ with $k = \inf\{\langle \tilde{d}, x\rangle | x \in \mathcal{C}\}$

Figure 6.1.: **Main idea of the lattice membership algorithm.** If $\tilde{d}$ is a non-zero integer vector, every integer vector in the convex set $\mathcal{C}$ is contained in an affine hyperplane $H_{k,\tilde{d}}$, where $k \in \mathbb{Z}$ with $\inf\{\langle \tilde{d}, x\rangle | x \in \mathcal{C}\} \leq k \leq \sup\{\tilde{d}, x\rangle | x \in \mathcal{C}\}$.

Definition 6.1.2. *(γ-flatness direction)*
Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a bounded convex set of dimension m. A vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ is called a γ-flatness direction *of $\mathcal{C}$ for some parameter $\gamma > 0$ if there exists an interval $I_{\mathcal{C}}$ of length at most γ such that the following holds: The convex set $\mathcal{C}$ contains an integer vector if and only if there exists $k \in \mathbb{Z} \cap I_{\mathcal{C}}$ such that $\mathcal{C} \cap H_{k,\tilde{d}}$ contains an integer vector.*

The parameter $\gamma > 0$ is arbitrary. It can be a constant or a function of any parameter associated to the lattice. Often the parameter depends on the dimension of the convex set.

In the following, if we say that we compute a γ-flatness direction $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ of some given convex set $\mathcal{C}$ together with a corresponding interval $I_{\mathcal{C}}$ we mean that we compute a vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ and an interval $I_{\mathcal{C}}$ of length at most γ such that the following holds: The convex set $\mathcal{C}$ contains an integer vector if and only if there exists $k \in \mathbb{Z} \cap I_{\mathcal{C}}$ such that $\mathcal{C} \cap H_{k,\tilde{d}}$ contains an integer vector.

At the moment, we assume that such a γ-flatness direction of a convex set can be found. Then we obtain a prototype of a lattice membership algorithm, which is described in Algorithm 6.

6.1.2. Description of the lattice membership algorithm

If we realize this idea of a membership algorithm, we obtain a recursive algorithm, where the recursive instances are given by a full-dimensional bounded convex set $\mathcal{C}$ and an affine subspace H. At the beginning, i.e., if $m = n$, we set $H := \mathbb{R}^n$. Later, the affine subspace H is given by a set of affine hyperplanes H_{k_i,d_i}, $m+1 \leq i \leq n$ for some parameter $m \leq n$.

Algorithm 6 Prototype of a lattice membership algorithm

```
Input: A full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ from the class $\mathcal{K}$ which is closed
    under bijective linear transformation.
Output: An integer vector in $\mathcal{C}$ or the statement that $\mathcal{C}$ does not contain an integer
    vector.
  If $n = 1$, find an integer vector in $\mathcal{C}$ or decide that $\mathcal{C}$ does not contain an integer vector.
  Otherwise,
      1. compute an $f(n)$-flatness direction $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ and an interval $I_{\mathcal{C}}$.
      2. For all $k \in \mathbb{Z} \cap I_{\mathcal{C}}$, find an integer vector in $\mathcal{C} \cap H_{k,\tilde{d}}$ or show that $\mathcal{C} \cap H_{k,\tilde{d}}$
         does not contain an integer vector.
      3. If there exists an index $k$ such that $\mathcal{C} \cap H_{k,\tilde{d}}$ contains an integer vector,
         output this vector. Otherwise, output that $\mathcal{C} \cap H_{k,\tilde{d}}$ does not contain an
         integer vector.
```

If $m = 0$, the affine subspace consists of a single vector. This vector can be computed efficiently using Gaussian elimination.

During the execution of the algorithm we need to be able to compute $f(m)$-flatness directions for the recursive instances. These recursive instances consist of a bounded convex set of dimension m given as the intersection $\mathcal{C} \cap H$ of the original input convex set $\mathcal{C}$ from the class $\mathcal{K}$ and an affine subspace H of dimension m. Thus, we assume that we have access to a so-called flatness algorithm $\mathcal{A}_{\mathcal{K},f}$.

Assumption 6.1.3. *(Existence of flatness algorithm for the class $\mathcal{K}$)*
Let $\mathcal{K}$ be a class of bounded convex sets and $f : \mathbb{N} \to \mathbb{R}^{>0}$ be some nondecreasing function. We assume that there exists a deterministic flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ that on input of a convex set $\mathcal{C} \in \mathcal{K}$ of dimension n together with an affine subspace H of dimension m computes an $f(m)$-flatness direction $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ of the convex set $\mathcal{C} \cap H$ together with a corresponding interval $I_{\mathcal{C} \cap H}$ of length at most $f(m)$.

In Section 6.4 we will show that for concrete classes of bounded convex sets, we can realize a flatness algorithm, in particular for polytopes and generalizations of ℓ_p-balls.

Under the assumption that we have access to a flatness algorithm satisfying Assumption 6.1.3, we are able to present a complete description of the algorithm, see Algorithm 7.

Theorem 6.1.4. *Let $\mathcal{K}$ be a class of bounded convex sets closed under bijective linear transformation. Assume that there exists a flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ satisfying Assumption 6.1.3.*

Algorithm 7 Membership algorithm for bounded convex sets

Input:

- A full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ from the class $\mathcal{K}$ which is closed under bijective linear transformation and satisfies Assumption 6.1.3, and
- an affine subspace $H := \bigcap_{i=m+1}^{n} H_{k_i,d_i}$, where $d_i \in \mathbb{Z}^n$ linearly independent and $k_i \in \mathbb{Z}$ for all $m+1 \leq i \leq n$; alternatively, $H := \mathbb{R}^n$.

Used subroutine: Flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ satisfying Assumption 6.1.3.

Output: An integer vector in $\mathcal{C} \cap H$ or the statement that $\mathcal{C} \cap H$ does not contain an integer vector.

If $m = 0$, compute a vector $z \in \mathbb{Z}^n \cap H$ satisfying $z \in \mathcal{C}$ or decide that $\mathcal{C} \cap H$ does not contain an integer vector.

Otherwise,

1. apply the flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ with input of the convex set $\mathcal{C}$ and the affine subspace H. The result is a vector $d_m \in \mathbb{Z}^n$ together with an interval $I_{\mathcal{C}\cap H}$.
2. For all $k \in \mathbb{Z} \cap I_{\mathcal{C}\cap H}$, apply the membership algorithm to the convex set $\mathcal{C}$ and the affine subspace $H \cap H_{k,d_m}$. Either the algorithm outputs an integer vector or it outputs that $\mathcal{C} \cap H \cap H_{k,d_m}$ does not contain an integer vector.
3. If there exists an index $k \in \mathbb{Z} \cap I_{\mathcal{C}\cap H}$ such that the algorithm outputs an integer vector, output this vector. Otherwise, output that $\mathcal{C} \cap H$ does not contain an integer vector.

Given a full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ from the class $\mathcal{K}$ and an affine subspace H, the lattice membership algorithm for bounded convex sets, Algorithm 7, decides correctly whether $\mathcal{C} \cap H$ contains an integer vector. If $\mathcal{C} \cap H$ contains an integer vector, it outputs one. The number of recursive calls of the algorithm is at most $(2f(m))^m$, where m is the dimension of the subspace.

Given a full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ and the whole vector space $\mathbb{R}^n$ as input, the algorithm solves the lattice membership problem correctly.

Proof. If $m = 0$, the affine subspace H consists of a single vector. Thus, the algorithm can decide correctly whether this vector is an integer vector which is contained in $\mathcal{C}$.
For $m \geq 1$, the membership algorithm applies the flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ to the full-dimensional bounded convex set $\mathcal{C}$ and the affine subspace H. By assumption, the algorithm computes an $f(m)$-flatness direction of the convex set $\mathcal{C} \cap H$ given by a vector $d_m \in \mathbb{Z}^n \backslash \{0\}$ and an interval $I_{\mathcal{C} \cap H}$. Since we assume that the flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ verifies Assumption 6.1.3, it is guaranteed that $\mathcal{C} \cap H$ contains an integer vector if and only if there exists an index $k \in \mathbb{Z} \cap I_{\mathcal{C} \cap H}$ such that $\mathcal{C} \cap H \cap H_{k,d_m}$ contains an integer vector.
Hence, if there exists an index $k \in \mathbb{Z} \cap I_{\mathcal{C} \cap H}$ such that the lattice membership algorithm with input of the convex set $\mathcal{C}$ and the affine subspace $H \cap H_{k,d_m}$ outputs an integer vector $v \in \mathcal{C} \cap H \cap H_{k,d_m}$, then we have found an integer vector in $\mathcal{C} \cap H$. Otherwise, i.e., if for all $k \in \mathbb{Z} \cap I_{\mathcal{C} \cap H}$ the set $\mathcal{C} \cap H \cap H_{k,d_m}$ does not contain an integer vector, it is guaranteed by Assumption 6.1.3 that $\mathcal{C} \cap H$ does not contain an integer vector.

If we are given a convex set in $\mathbb{R}^n$ together with an affine subspace of dimension m as input, we need at most $f(m) + 1$ solutions of recursive instances where the dimension of the subspace is $m - 1$, since the length of the interval $I_{\tau(\mathcal{C} \cap H)}$ is at most $f(m)$. Hence, the overall number of recursive calls is at most

$$\prod_{i=1}^{m} (f(i) + 1) \leq 2^m f(m)^m.$$

□

In the next proposition we show that our lattice membership algorithm runs in polynomial space if the bit size of each number computed by the algorithm is polynomial in the bit size of the input instance. So far we do not specify how the convex sets from the class $\mathcal{K}$ are given. Hence, the notion of the size of a convex set $\mathcal{C}$ from the class $\mathcal{K}$ is not clearly defined, but this does not matter in the following statement.

Proposition 6.1.5. *Let $\mathcal{K}$ be a class of bounded convex sets closed under bijective linear transformation. Assume that there exists a flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ satisfying Assumption 6.1.3. Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional convex set from the class $\mathcal{K}$ and let $H \subseteq \mathbb{R}^n$ be an affine subspace of dimension m. Let r be an upper bound on the size of $\mathcal{C}$ and H. If the flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ runs in polynomial space and if all numbers computed by the lattice membership algorithm, Algorithm 7, with input of the convex set $\mathcal{C} \subseteq \mathbb{R}^n$ and the* *affine subspace H have size at most r^{n^c} for some fixed constant $c > 1$, then the lattice membership algorithm runs in polynomial space.*

Proof. The lattice membership algorithm with input of an affine subspace H of dimension m is a recursive algorithm where the corresponding recursion tree has m levels. Each level consists of all recursive instances with the corresponding affine subspaces having the same dimension k.
Let $s(k)$ be an upper bound on the space used by the lattice membership algorithm given as input an affine subspace of dimension k. By assumption, the size of each number computed in one reduction step is at most r^{n^c}. Since the flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ runs in polynomial space, this shows that this reduction step runs in polynomial space, i.e., it needs space at most $(r^{n^c})^{\mathcal{O}(1)}$.
At each step in our lattice membership algorithm we need to consider exactly one path in the recursion tree. Furthermore, the algorithm can terminate if it has found an integer vector in $\mathcal{C} \cap H$. The recursive instances are given by a vector $d_k \in \mathbb{Z}^n \backslash \{0\}$ together with an interval $I = [k_{\min}, k_{\max}]$, which is given by its lower and upper bound. By assumption, the size of all these numbers is at most r^{n^c}. This shows that the space complexity satisfies the following recursion

$$s(k) \leq \mathcal{O}(r^{n^c}) + s(k-1).$$

From this, it follows that the space complexity of the algorithm is upper bounded by $\sum_{k=0}^{m} \mathcal{O}(r^{n^c}) = r^{n^{\mathcal{O}(1)}}$, that means the algorithm runs in polynomial space. □

Unfortunately, for the outline of our lattice membership algorithm presented so far we cannot guarantee that the bit size of each number computed by the algorithm is polynomial in the bit size of the input instance. In fact, the size of the new affine hyperplane depends not only on the size of the convex set $\mathcal{C}$ but also on the size of the affine subspace. This problem occurs also in Lenstra's algorithm for integer programming and its improvement by Kannan. To avoid this problem, we use a replacement procedure developed by Frank and Tardos in 1987, see [FT87]. In the next section, we will describe their result and show how it can be used to obtain a polynomially space bounded algorithm for the lattice membership problem.

6.1.3. A polynomially space bounded lattice membership algorithm

The replacement procedure from Frank and Tardos is a polynomial time algorithm that on input of an affine subspace $H \subseteq \mathbb{R}^n$ and an additional hyperplane $H_{k,d}$ computes a set of new affine hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$ for some index set J with small size.

If the affine subspace H and the affine hyperplane $H_{k,d}$ are affinely independent, the affine subspace H and the new affine hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$ are also affinely independent and we have $\dim(H) + |J| \leq n$. Furthermore, if we additionally consider a convex set $\mathcal{C}$ and choose the parameters appropriately depending on the shape of the convex set, it can be guaranteed that each integer vector in the convex set $\mathcal{C}$ is contained in the affine subspace $H \cap H_{k,d}$ if and only if it is contained in the intersection $H \cap \bigcap_{i \in J} H_{\bar{k}_i, \bar{d}_i}$. The following result is a slight generalization of Lemma 5.1 in [FT87]. Its proof together with a short description of the procedure appears in Section 6.5 at the end of this chapter.

Proposition 6.1.6. *There exists a replacement procedure, which satisfies the following properties: Given a parameter $N \in \mathbb{N}$ as input as well as an affine subspace $H \subseteq \mathbb{R}^n$ and an additional affine hyperplane $H_{k,d}$ which is affinely independent of H, the replacement procedure computes a set of affinely independent hyperplanes $H_{\bar{k}_i, \bar{d}_i}$, $i \in J \neq \emptyset$ such that the following holds:*

- *Every integer vector $z \in \bar{B}_n^{(1)}(0, N-1) \cap H$ satisfies $\langle d, z \rangle = k$ if and only if it satisfies $\langle \bar{d}_i, z \rangle = \bar{k}_i$ for all $i \in J$.*
- *The affine subspace H and the affine hyperplane $H_{\bar{k}_i, \bar{d}_i}$, $i \in J$ are affinely independent.*

The size of the vectors $\bar{d}_i \in \mathbb{Z}^n$ and the numbers $\bar{k}_i \in \mathbb{Z}$ is at most $2^{(n+2)^2} N^n$. The number of arithmetic operations of the replacement procedure is at most $(n \cdot \log_2(N))^{\mathcal{O}(1)}$.

If we use this replacement procedure in the lattice membership algorithm with a suitable computed parameter N directly before the recursive call of the lattice membership algorithm, we can replace the newly constructed affine hyperplane H_{k,d_m} with the intersection of several other affine hyperplanes whose size depend only on the shape of the convex set $\mathcal{C}$ and not on the size of the affine subspace H. The parameter N depends only on the shape of the convex set, that means in each recursion step of the lattice membership algorithm we can use the same parameter N. Hence, if we describe our membership algorithm and say that we compute in each recursion step a parameter N such that $\mathcal{C} \subseteq \bar{B}_n^{(1)}(0, N-1)$ this is only for the simplification of the representation. A complete description of this algorithm, which is called the modified membership algorithm, is given in Algorithm 8.

Theorem 6.1.7. *Let $\mathcal{K}$ be a class of bounded convex sets closed under bijective linear transformation. Assume that there exists a flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ satisfying Assumption 6.1.3.*
Given a full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ from the class $\mathcal{K}$ satisfying $\mathcal{C} \subseteq \bar{B}_n^{(1)}(0, N-1)$ and an affine subspace H, the modified lattice membership algorithm, Algorithm 8, satisfies the following properties: It decides correctly whether $\mathcal{C} \cap H$ contains an integer vector or not. If $\mathcal{C} \cap H$ contains an integer vector, it outputs one.
Each recursive instance consists of the original convex set $\mathcal{C}$ and an affine subspace of size of at most

$$\max\left\{\text{size}(H), 2^{(n+2)^2} N^n\right\}.$$

Proof. Since $\mathcal{C} \subseteq \bar{B}_n^{(1)}(0, N-1)$, for all $k \in \mathbb{Z}$ the convex set $\mathcal{C}$ contains an integer vector from $H \cap H_{k,d_m}$, if and only if it contains an integer vector from $H \cap \bigcap_{i \in J_k} H_{\bar{k}_i, \bar{d}_i}$, see Proposition 6.1.6. Hence, the correctness of the algorithm follows directly from Theorem

Algorithm 8 A polynomially space bounded lattice membership algorithm for bounded convex sets

Input:

- A full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ from the class $\mathcal{K}$, which is closed under bijective linear transformation and satisfies Assumption 6.1.3, and
- an affine subspace $H := \bigcap_{i=m+1}^{n} H_{k_i,d_i}$, where $d_i \in \mathbb{Z}^n$ linearly independent and $k_i \in \mathbb{Z}$ for all $m+1 \leq i \leq n$; alternatively, $H := \mathbb{R}^n$.

Used subroutines: Flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ satisfying Assumption 6.1.3; replacement procedure.

Output: An integer vector in $\mathcal{C} \cap H$ or the statement that $\mathcal{C} \cap H$ does not contain an integer vector.

If $m = 0$, compute a vector $z \in \mathbb{Z}^n \cap H$ satisfying $z \in \mathcal{C}$ or decide that $\mathcal{C} \cap H$ does not contain an integer vector.

Otherwise,

1. apply the flatness algorithm $\mathcal{A}_{\mathcal{K},f}$ with input of the convex set $\mathcal{C}$ and the affine subspace H. The result is a vector $d_m \in \mathbb{Z}^n$ together with an interval $I_{\mathcal{C} \cap H}$.
2. Compute a parameter $N \in \mathbb{N}$ with $\mathcal{C} \subseteq \bar{B}_n^{(1)}(0, N-1)$.
3. For all $k \in \mathbb{Z} \cap I_{\mathcal{C}}$,
   a) apply the replacement procedure to the affine subspace H, the hyperplane H_{k,d_m} and the parameter N. The result is an index set J_k and an affine subspace $\bigcap_{i \in J_k} H_{\bar{k}_i,\bar{d}_i}$.
   b) Apply the membership algorithm to the convex set $\mathcal{C}$ and the affine subspace $H \cap \bigcap_{i \in J_k} H_{\bar{k}_i,\bar{d}_i}$. Either the algorithm outputs an integer vector or it outputs that $\mathcal{C} \cap H \cap \bigcap_{i \in J_k} H_{\bar{k}_i,\bar{d}_i}$ does not contain an integer vector.
4. If there exists an index k such that the algorithm outputs an integer vector, output this vector. Otherwise, output that $\mathcal{C} \cap H$ does not contain an integer vector.

6.1.4. Each recursive instance consists of the original convex set $\mathcal{C}$ and an affine subspace of dimension $m-1$. This subspace is given as the intersection of the original subspace H and an affine hyperplane of size of at most $2^{(n+2)^2}N^n$, see again Proposition 6.1.6. Hence, the size of the recursive affine subspace is the maximum of the size of the affine subspace H and $2^{(n+2)^2}N^n$. □

Obviously, we are able to adapt this general framework for all classes of bounded convex sets for which there exists a flatness algorithm. In the following two sections, we consider polytopes and generalizations of ℓ_p-balls. We will see that for these classes of bounded convex sets, there exists a flatness algorithm.

6.2. A lattice membership algorithm for polytopes

In this section, we consider the class of full-dimensional polytopes. We will present a deterministic algorithm that solves the lattice membership problem for these convex sets. Given a full-dimensional polytope in $\mathbb{R}^n$ defined by s constraints as input, the running time of this algorithm is $(s \cdot \log_2(r))^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r is an upper bound on the size of the input polytope.

In the following, we always assume that the polytopes are given by a matrix $A \in \mathbb{Z}^{s\times n}$ and a vector $\beta \in \mathbb{Z}^s$. Obviously, the class of all full-dimensional polytopes is closed under bijective linear transformation. Furthermore, there exists a flatness algorithm for polytopes. The proof of the following result together with a description of the algorithm appears in Section 6.4 at the end of this chapter.

Theorem 6.2.1. *There exists a flatness algorithm that for all full-dimensional polytopes $P \subseteq \mathbb{R}^n$ and affine subspaces H of dimension m outputs a $2m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $P \cap H$ together with a corresponding interval $I_{P\cap H} \subseteq \mathbb{R}$ of length of at most $2m^2$. The number of arithmetic operations of the flatness algorithm is*

$$(ns \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} m^{m/(2e)},$$

where r is an upper bound on the size of the polytope, s is the number of constraints defining the polytope, and e is Euler's constant. The algorithm runs in polynomial space and each number computed by the algorithm has size of at most $r^{n^{\mathcal{O}(1)}}$.

Using this result, we can adapt the algorithmic framework from Section 6.1 to solve the lattice membership problem for polytopes. Essentially, the lattice membership algorithm for polytopes works in the same way as the lattice membership algorithm for bounded convex sets as presented in Algorithm 8. As input, the algorithm gets a full-dimensional polytope in $\mathbb{R}^n$ given by integral constraints and an affine subspace $H \subseteq \mathbb{R}^n$. Furthermore, we assume that the polytope is given together with an upper bound on its size $r_P = \text{size}(P)$.

Algorithm 9 Lattice membership algorithm for polytopes

Input:

- A full-dimensional polytope P given by $A \in \mathbb{Z}^{s \times n}$ and $\beta \in \mathbb{Z}^s$ together with its size r_P and
- an affine subspace $H := \bigcap_{i=m+1}^{n} H_{k_i,d_i}$ given by $d_i \in \mathbb{Z}^n$ linearly independent and $k_i \in \mathbb{Z}$, $m+1 \leq i \leq n$; alternatively, $H := \mathbb{R}^n$.

Used subroutines: Flatness algorithm, replacement procedure.

Output: An integer vector in $P \cap H$ or the statement that $P \cap H$ does not contain an integer vector.

If $m = 0$, compute a vector $z \in \mathbb{Z}^n \cap H$ satisfying $z \in P$ or decide that $P \cap H$ does not contain an integer vector.

Otherwise,

1. apply the flatness algorithm to the polytope P and the affine subspace H. The result is a vector $d_m \in \mathbb{Z}^n$ together with an interval $I_{P \cap H} \subseteq \mathbb{R}$.
2. Set $N \leftarrow n^{(n+3)/2} r_P^n + 1$.
3. For all $k \in \mathbb{Z} \cap I_{P \cap H}$,
   a) apply the replacement procedure to the affine subspace H, the hyperplane H_{k,d_m} and the parameter N.
   The result is an index set J_k and an affine subspace $\bigcap_{i \in J_k} H_{\bar{k}_i,\bar{d}_i}$.
   b) Apply the membership algorithm to the polytope P and the affine subspace $H \cap \bigcap_{i \in J_k} H_{\bar{k}_i,\bar{d}_i}$.
   Either the algorithm outputs an integer vector or it outputs that $P \cap H \cap \bigcap_{i \in J_k} H_{\bar{k}_i,\bar{d}_i}$ does not contain any integer vector.
4. If there exists an index k such that the algorithm outputs an integer vector, output this vector. Otherwise, output that $P \cap H$ does not contain an integer vector.

In the first step, it applies the flatness algorithm to the full-dimensional polytope P and the affine subspace H. As a result, we obtain a $2m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $P \cap H$ together with an interval $I_{P \cap H}$ and we check recursively if there exists an integer $k \in I_{P \cap H}$ such that $P \cap H \cap H_{k,d_m}$ contains an integer vector.
For the computation of the parameter N, we use that the size of the vertices of every full-dimensional polytope given by integral constraints are at most $n^{(n+1)/2} r_P^n$ (in absolute value), where n is the dimension and r_P is the size of the corresponding polytope. Hence, we set N as $n^{(n+3)/2} r_P^n$. A detailed description of the algorithm is given in Algorithm 9.

In the next theorem, we show that the lattice membership algorithm for polytopes can be used to solve the lattice membership problem. To prove this we use the result for the lattice membership problem for bounded convex sets to show that on input of a full-dimensional polytope P and an affine subspace H the lattice membership algorithm for polytopes decides correctly whether the intersection of the polytope with the affine subspace contains an integer vector.

Theorem 6.2.2. *Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope given by a matrix $A \in \mathbb{Z}^{s \times n}$ and a vector $\beta \in \mathbb{Z}^s$. Let $H \subseteq \mathbb{R}^n$ be an affine subspace of dimension $m \leq n$.*
Given P and H as input, the lattice membership algorithm for polytopes, Algorithm 9, finds an integer vector in $P \cap H$ if there exists one. Otherwise, it outputs that $P \cap H$ does not contain an integer vector. The number of arithmetic operations of the algorithm is

$$(n \cdot s \log_2(r))^{\mathcal{O}(1)} m^{(2+o(1))m},$$

where r is an upper bound on the size of the polytope and the affine subspace. The algorithm runs in polynomial space and each number computed by the algorithm has size of at most $r^{n^{\mathcal{O}(1)}}$.

Proof. Since P is a full-dimensional polytope given by integral constraints, P is contained in an ℓ_∞-ball with radius $n^{(n+1)/2} r_P^n$, where r_P is an upper bound on the size of the polytope, see Lemma 2.2.19 in Chapter 2. According to Hölder's inequality, it follows that

$$P \subseteq \bar{B}_n^{(1)}(0, n^{(n+3)/2} r_P^n).$$

This shows that the parameter N computed by the algorithm satisfies $P \subseteq \bar{B}_n^{(1)}(0, N-1)$. Combining this with Theorem 6.1.7, it follows that the lattice membership algorithm for polytopes decides correctly whether $P \cap H$ contains an integer vector or not. If $P \cap H$ contains an integer vector, it outputs such a vector.

Now we consider the size of the numbers computed by the lattice membership algorithm for polytopes. We assume that we are given as input a full-dimensional polytope P of size r_P and an affine subspace $H \subseteq \mathbb{R}^n$ of dimension m. The parameter r is an upper bound on the size of both, i.e., $r \geq \max\{r_P, \text{size}(H)\}$.

The primary observation is that each recursive instance of the lattice membership algorithm for polytopes consists of the original input polytope P together with an affine subspace $\tilde{H}$ of size of at most

$$\max\left\{\text{size}(H), 2^{(n+2)^2} N^n\right\}, \tag{6.1}$$

see Theorem 6.1.7. By definition, the parameter N is at most

$$N = n^{(n+3)/2} r_P^n + 1 \leq r^{n^{\mathcal{O}(1)}}. \tag{6.2}$$

It is important that the parameter N depends only on the size of the polytope and not on the size of the affine subspace. This shows that each recursive instance of the lattice membership algorithm consists of an affine subspace of size of at most $r^{n^{\mathcal{O}(1)}}$.

In the following, we consider an arbitrary recursive instance of the lattice membership algorithm given by the polytope P and an affine subspace $\tilde{H}$ of dimension $k \leq m$.
If we apply the flatness algorithm to the polytope P and the affine subspace $\tilde{H}$, each number computed by the flatness algorithm, in particular the bounds of the interval $I_{P\cap\tilde{H}}$ and the vector d_k, has size of at most

$$\max\left\{r_P, \operatorname{size}(\tilde{H})\right\}^{n^{\mathcal{O}(1)}} = r^{n^{\mathcal{O}(1)}},$$

see Theorem 6.2.1. Also, the other numbers computed by the membership algorithm in one recursion step have size of at most $r^{n^{\mathcal{O}(1)}}$.

Since the size of the affine subspace $\tilde{H}$ is at most $r^{n^{\mathcal{O}(1)}}$, this shows that all numbers computed by the lattice membership algorithm in one reduction step have size $r^{n^{\mathcal{O}(1)}}$. Overall, this shows that the size of each number computed by the lattice membership algorithm is at most $r^{n^{\mathcal{O}(1)}}$ and that the algorithm runs in polynomial space, see Proposition 6.1.5.

Finally, we give an upper bound on $T(m, n, s, r)$, the number of arithmetic operations of the lattice membership algorithm for polytopes with input of a full-dimensional polytope in $\mathbb{R}^n$ given by a matrix $A \in \mathbb{Z}^{s\times n}$ and a vector $\beta \in \mathbb{Z}^s$ and an affine subspace H of dimension m. The parameter r is an upper bound on the size of the polytope and the affine subspace.

As already said, the recursive instances of the algorithm consist of the original input polytope P and an affine subspace $\tilde{H}$ of dimension k with $0 \leq k \leq m$.
We start with the case $k = 0$, which means that the affine subspace H consists of a single vector. This vector can be computed using Gaussian elimination in $\mathcal{O}(n^3)$ arithmetic operations. Using $\mathcal{O}(s)$ arithmetic operation, it can be checked if this vector is contained in the polytope, i.e.,

$$T(0, n, s, r) = (s \cdot n)^{\mathcal{O}(1)}.$$

We now assume that the dimension of the affine subspace $\tilde{H}$ is $k > 0$. The number of arithmetic operations of the flatness algorithm is at most

$$s \cdot n \cdot \log_2\left(\max\{\operatorname{size}(P), \operatorname{size}(\tilde{H})\}\right) 2^{\mathcal{O}(k)} k^{k/(2e)},$$

see Theorem 6.2.1. As we have seen above, the size of the affine subspace $\tilde{H}$ is at most $r^{n^{\mathcal{O}(1)}}$, which shows that the number of arithmetic operations of the flatness algorithm is at most

$$(s \cdot n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(k)} k^{k/(2e)}.$$

It is particularly important that the number of arithmetic operations depends only on the size of the polytope and the input subspace H and not on the size of the affine subspace $\tilde{H}$ of the recursive instance.
The number of arithmetic operations of the replacement procedure is polynomial in n and $\log_2(N)$, see Proposition 6.1.6. By our definition of N, we have $\log_2(N) \leq (n+3)\log_2(n \cdot r)$. This shows that the number of arithmetic operations of the replacement procedure is at most $(n \cdot \log_2(r))^{\mathcal{O}(1)}$. Overall, this shows that in one recursive instance the number of arithmetic operations is at most

$$(s \cdot n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(k)} k^{k/(2e)},$$

where k is the dimension of the affine subspace defining the recursive instance.

The number of recursive calls of the lattice membership algorithm is determined by the length of the interval computed by the flatness algorithm. The length of this interval is at most $2k^2$, see Theorem 6.2.1. Thus, there exist at most

$$2m^2 \cdot 2(m-1)^2 \cdot \ldots \cdot 2(m-k+1)^2 = 2^{m-k}\left(\frac{m!}{k!}\right)^2$$

different recursive instances where the corresponding affine subspace has dimension k. Hence, the number of arithmetic operations is upper bounded by

$$\sum_{k=0}^{m} (s \cdot n \cdot \log_2(r))^{\mathcal{O}(1)}\, 2^{\mathcal{O}(k)} k^{k/(2e)} 2^{m-k}\left(\frac{m!}{k!}\right)^2.$$

That means there exist constants $c_1, c_2 \geq 1$ such that the number of arithmetic operations of the membership algorithm is at most

$$\begin{aligned} T(m,n,s,r) &\leq \sum_{k=0}^{m} (s \cdot n \cdot \log_2(r))^{c_1}\, 2^{c_2 \cdot k} k^{k/(2e)} 2^{m-k}\left(\frac{m!}{k!}\right)^2 \\ &\leq (s \cdot n \cdot \log_2(r))^{c_1}\, 2^{(c_2+1)m} \sum_{k=0}^{m} k^{k/(2e)}\left(\frac{m!}{k!}\right)^2 \end{aligned}$$

Using Stirling's formula, we see that $m! \leq m^m$ and that $k! \geq (k/e)^k$. Thus, the number of arithmetic operations can be upper bounded by

$$\begin{aligned} T(m,n,s,r) &\leq (s \cdot n \cdot \log_2(r))^{c_1} 2^{(c_2+1)m} \sum_{k=0}^{m} k^{k/(2e)} m^{2m} k^{-2k} e^{2k} \\ &\leq (s \cdot n \cdot \log_2(r))^{c_1} 2^{(c_2+1)m} m^{2m} e^{2m} \sum_{k=0}^{m} k^{(1/(2e)-2)k} \\ &= (s \cdot n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} m^{2m}. \end{aligned}$$

□

If we apply the membership algorithm with input of a full-dimensional polytope and the vector space we obtain an algorithm for the lattice membership problem.

Corollary 6.2.3. *The lattice membership algorithm for polytopes, Algorithm 9, solves the lattice membership problem for all full-dimensional polytopes given by a matrix $A \in \mathbb{Z}^{s\times n}$ and a vector $\beta \in \mathbb{Z}^s$ correctly. The number of arithmetic operations of the algorithm is at most $s^{\mathcal{O}(1)} \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r is an upper bound on the size of the polytope. The algorithm runs in polynomial space and each number produced by the algorithm has size of at most $r^{n^{\mathcal{O}(1)}}$, that means bit size of at most $n^{\mathcal{O}(1)} \log_2(r)$.*

6.3. A lattice membership algorithm for ℓ_p-balls

In this section, we use the algorithmic framework presented in Section 6.1 to obtain an algorithm that solves the lattice membership problem for ℓ_p-balls with $1 < p < \infty$.

Unfortunately, the class of ℓ_p-balls is not closed under linear affine transformation. Hence, we consider generalizations of ℓ_p-balls, so called general ℓ_p-balls, in this section. We will present a polynomially space bounded lattice membership algorithm for general ℓ_p-balls. The number of arithmetic operations of this algorithm is $p \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r is an upper bound on the size of the general ℓ_p-ball and n is its dimension. Obviously, we also obtain an algorithm which solves the lattice membership problem for ℓ_p-balls. Before we describe this algorithm, we will define the class of ℓ_p-balls.

6.3.1. The class of general ℓ_p-balls

General ℓ_p-balls are balls generated by the generalization of an ℓ_p-norm. The generalization of an ℓ_p-norm, $1 < p < \infty$, is defined in the same way as the generalization of the Euclidean norm, see Section 2.2.1. For the definition of general ℓ_p-balls we consider generalizations of ℓ_p-norms. By the generalization of an ℓ_p-norm with $1 < p < \infty$, we understand the following: We consider norms, whose unit balls are transformations of the ℓ_p-unit ball.

Definition 6.3.1. *Let $1 < p < \infty$ and $V \in \mathbb{R}^{n\times n}$ be nonsingular. For a vector $x \in \mathbb{R}^n$ we define*

$$\|x\|_p^V := \|V^{-1}x\|_p.$$

Obviously, the mapping $\|\cdot\|_p^V$ defines a norm on $\mathbb{R}^n$. If the matrix V is an orthogonal matrix, the unit ball of this norm is just the rotation of the ℓ_p-unit ball by the orthogonal matrix V. For an illustration of a generalized ℓ_p-ball see Figure 6.2.

Like all convex functions, the generalized ℓ_p-norms can be used to define convex sets. Given a nonsingular matrix $V \in \mathbb{R}^{n\times n}$ together with a vector $t \in \mathbb{R}^n$ and a parameter $\alpha > 0$, we define the set $B_n^{(p,V)}(t,\alpha)$ as the set of all vectors in $\mathbb{R}^n$ whose distance to t with respect to the norm $\|\cdot\|_p^V$ is at most α,

$$B_n^{(p,V)}(t,\alpha) := \{x \in \mathbb{R}^n | \|x-t\|_p^V < \alpha\}.$$

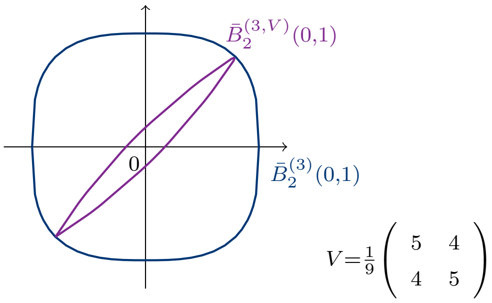

Figure 6.2.: **General ℓ_p-balls.** In this picture, we see the unit-ball of the ℓ_3-norm together with the unit ball of the norm $\|\cdot\|_3^V$.

Analogously, $\bar{B}_n^{(p,V)}(t,\alpha) = \{x \in \mathbb{R}^n | \|x-t\|_p^V \leq \alpha\}$. If we consider the standard ℓ_p-norm, we omit the matrix I_n and write $B_n^{(p)}(t,\alpha)$ instead.

Obviously, the class of all sets $B_n^{(p,V)}(t,\alpha)$ is closed under bijective affine transformation. In the following, whenever we speak of a general ℓ_p-ball, we assume that we are given a nonsingular matrix $V \in \mathbb{R}^{n\times n}$, a vector $t \in \mathbb{R}^n$, and a parameter $\alpha > 0$ and we consider the convex set $B_n^{(p,V)}(t,\alpha)$. The size of such a general ℓ_p-ball is the maximum of n, α and the size of the coordinates of V^{-1} and t.

In the following, we will present a deterministic algorithm that solves the lattice membership problem for general ℓ_p-balls, $1 < p < \infty$.

6.3.2. Description and analysis of the algorithm

In Section 6.4.3, we will show that for all general ℓ_p-balls there exists a flatness algorithm. The flatness algorithm for general ℓ_p-balls differs from the flatness algorithm for polytopes in the point that it is possible that the flatness algorithm outputs that the general ℓ_p-ball does not contain an integer vector. Obviously, this is not a problem in our setting.

Theorem 6.3.2. *There exists a flatness algorithm that given a general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ with $1 < p < \infty$ together with an affine subspace H of dimension m outputs one of the following:*

- *Either it outputs that $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector, or*
- *it outputs a $4m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $B_n^{(p,V)}(t,\alpha) \cap H$ together with a corresponding interval $I_{\mathcal{B}\cap H}$ of length at most $4m^2$.*

The number of arithmetic operations of the algorithm is

$$p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} m^{m/(2e)},$$

where r is an upper bound on the size of the general ℓ_p-ball and e is Euler's constant. The algorithm runs in polynomial space and each number computed by the algorithm has size of at most $r^{pn^{\mathcal{O}(1)}}$,

Using this algorithm, we are able to describe an algorithm that solves the lattice membership problem for the class of general ℓ_p-balls with $1 < p < \infty$. In particular, we obtain an algorithm that solves the lattice membership problem for ℓ_p-balls.

Substantially, the algorithm works in the same way as the lattice membership algorithm for bounded convex sets presented in Section 6.1. The algorithm gets as input a full-dimensional general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ together with an upper bound on its size $r_{\mathcal{B}}$ and an affine subspace H of dimension m. Either it outputs an integer vector in $B_n^{(p,V)}(t,\alpha) \cap H$ or it outputs that $B_n^{(p,V)}(t,\alpha)$ does not contain an integer vector.

As in the general lattice membership algorithm, the lattice membership algorithm for general ℓ_p-balls applies the flatness algorithm with input of the general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ and the affine subspace H in the first step. If the flatness algorithm outputs that the set $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector, the membership algorithm outputs the same. Otherwise, the flatness algorithm outputs a $4m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $B_n^{(p,V)}(t,\alpha) \cap H$ together with a corresponding interval $I_{\mathcal{B}\cap H}$. In this case, the membership algorithm checks recursively whether there exists an integer $k \in I_{\mathcal{B}\cap H}$ such that $B_n^{(p,V)}(t,\alpha) \cap H \cap H_{k,d_m}$ contains an integer vector.

To apply the replacement procedure, we need to be able to compute the radius of a circumscribed ℓ_1-ball for a given general ℓ_p-ball. For this, we use the following observation.

Lemma 6.3.3. *Let $B_n^{(p,V)}(t,\alpha)$ be a general ℓ_p-ball given by $V \in \mathbb{R}^{n\times n}$ nonsingular, $t \in \mathbb{R}^n$, $\alpha > 0$ and $1 < p < \infty$. Then $B_n^{(p,V)}(t,\alpha)$ is contained in a Euclidean ball with radius $\alpha\sqrt{n}\|V\|_2$, where $\|V\|_2$ denotes the spectral norm of the matrix V.*

Proof. Using Hölder's inequality, we obtain that the ℓ_p-body $B_n^{(p,V)}(t,\alpha)$ is contained in the set $\{x \in \mathbb{R}^n | \|V^{-1}(x-t)\|_2 \leq \alpha\sqrt{n}\}$, which is the ellipsoid $\alpha\sqrt{n} \star E(VV^T,t)$. The circumscribed radius of an ellipsoid is given by the square root of the largest eigenvalue of the matrix defining it, see Lemma 2.2.10. The square root of the largest eigenvalue of VV^T is the spectral norm of the matrix V,

$$\|V\|_2 = \|V^T\|_2 = \max\left\{ \sqrt{\frac{x^T V V^T x}{x^T x}} \;\middle|\; x \in \mathbb{R}^n \backslash \{0\} \right\}.$$

Hence, we obtain that

$$B_n^{(p,V)}(t,\alpha) \subseteq \bar{B}_n^{(2)}(t, \alpha\sqrt{n}\|V\|_2).$$

□

This shows that a general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ is contained in an ℓ_1-ball with radius $2n \cdot r_{\mathcal{B}}\|V\|_2$, where $r_{\mathcal{B}}$ is an upper bound on the size of the general ℓ_p-ball. That means, we can define the parameter N as $n \cdot r_{\mathcal{B}}\|V\|_2 + 1$. A detailed description of the algorithm is given in Algorithm 10.

Theorem 6.3.4. *Let $B_n^{(p,V)}(t,\alpha)$ be a general ℓ_p-ball given by $V \in \mathbb{Q}^{n\times n}$ nonsingular, $t \in \mathbb{Q}^n$, $\alpha > 0$ and $1 < p < \infty$ and let H be an affine subspace of dimension $m \leq n$. Given as input $B_n^{(p,V)}(t,\alpha)$ and H, the membership algorithm for general ℓ_p-balls, Algorithm 10, decides correctly whether $B_n^{(p,V)}(t,\alpha) \cap H$ contains an integer vector. If there exists an integer vector in $B_n^{(p,V)}(t,\alpha) \cap H$, the algorithm outputs such a vector. The number of arithmetic operations of the algorithm is at most*

$$p(n\log_2(r))^{\mathcal{O}(1)} m^{(2+o(1))m},$$

where r is an upper bound on the size of $B_n^{(p,V)}(t,\alpha)$ and the size of H. The algorithm runs in polynomial space and each number computed by the algorithm has size at most $r^{p\cdot n^{\mathcal{O}(1)}}$.

Proof. Without loss of generality, we assume that the flatness algorithm with input of the general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ and the affine subspace H outputs a $4m^2$-flatness direction $d_m \in \mathbb{Z}^n\backslash\{0\}$ together with a corresponding interval $I_{\mathcal{B}\cap H}$. We have seen in Lemma 6.3.3 that $B_n^{(p,V)}(t,\alpha)$ is contained in a Euclidean ball with radius $\alpha\sqrt{n}\|V\|_2$. Using Hölder's inequality, this shows that

$$B_n^{(p,V)}(t,\alpha) \subseteq \bar{B}_n^{(2)}(t,\alpha\sqrt{n}\|V\|_2) \subseteq \bar{B}_n^{(1)}(0,nr_{\mathcal{B}}\|V\|_2),$$

since $r_{\mathcal{B}}$ is an upper bound on the size of the general ℓ_p-ball, i.e., $r_{\mathcal{B}} \geq \alpha$.
By definition of N this shows that $B_n^{(p,V)}(t,\alpha) \subseteq \bar{B}_n^{(1)}(0,N-1)$. Hence, if the flatness algorithm outputs a vector $d_m \in \mathbb{Z}^n\backslash\{0\}$ together with an interval $I_{\mathcal{B}\cap H}$, it follows from Theorem 6.1.7 that the membership algorithm for general ℓ_p-balls decides correctly whether $B_n^{(p,V)}(t,\alpha) \cap H$ contains an integer vector and outputs some if $B_n^{(p,V)}(t,\alpha) \cap H$ contains an integer vector.

Now, we consider the size of each number computed by the algorithm given a general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ of size $r_{\mathcal{B}}$ and an affine subspace $H \subseteq \mathbb{R}^n$ of dimension m as input. The parameter r is an upper bound on the size of both, i.e., $r \geq \max\{r_{\mathcal{B}}, \text{size}(H)\}$.

The primary observation is that each recursive instance consists of the original input ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ together with an affine subspace $\tilde{H}$ of size of at most

$$\max\{\text{size}(H), 2^{(n+2)^2}N^n\}, \tag{6.3}$$

see Theorem 6.1.7. Since the parameter N is at most $n \cdot r_{\mathcal{B}}\|V\|_2 + 1 \leq 2nr_{\mathcal{B}}\|V\|_2$, the size of the affine subspace is upper bounded by

$$2^{(n+2)^2}N^n \leq 2^{(n+2)^2}2^n n^n r_{\mathcal{B}}^n\|V\|_2^n \leq 2^{2(n+2)^2}(n\cdot r_{\mathcal{B}}\|V\|_2)^n \leq r_{\mathcal{B}}^{n^{\mathcal{O}(1)}}. \tag{6.4}$$

Algorithm 10 Lattice membership algorithm for general ℓ_p-balls, $1 < p < \infty$

```
Input:
  • A general $\ell_p$-ball $B_n^{(p,V)}(t,\alpha)$ given by a nonsingular matrix $V \in \mathbb{Q}^{n\times n}$, a vector
    $t \in \mathbb{Q}^n$ and a radius $\alpha > 0$ together with its size $r_\mathcal{B}$ and
  • an affine subspace $H := \bigcap_{i=m+1}^{n} H_{k_i,d_i}$ given by $d_i \in \mathbb{Z}^n$ linearly independent
    and $k_i \in \mathbb{Z}$, $m+1 \leq i \leq n$; alternatively, $H := \mathbb{R}^n$.

Used subroutines: Flatness algorithm, replacement procedure.

Output: An integer vector in $B_n^{(p,V)}(t,\alpha) \cap H$ or the statement that $B_n^{(p,V)}(t,\alpha) \cap H$
  does not contain an integer vector.

 If $m = 0$, compute a vector $z \in \mathbb{Z}^n \cap H$ satisfying $z \in B_n^{(p,V)}(t,\alpha)$ or decide that
    $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector.
  Otherwise, apply the flatness algorithm with input of the general $\ell_p$-ball $B_n^{(p,V)}(t,\alpha)$
    and the affine subspace $H$.
    If it outputs that $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector, then output
        the same.
    Otherwise, the result is a vector $d_m \in \mathbb{Z}^n \backslash \{0\}$ together with an interval $I_{\mathcal{B}\cap H}$.
        1. Set $N := n \cdot r_\mathcal{B} \cdot \|V\|_2 + 1$.
        2. For all $k \in \mathbb{Z} \cap I_{\mathcal{B}\cap H}$,
            a) apply the replacement procedure to the affine subspace $H$, the
               hyperplane $H_{k,d_m}$ and the parameter $N$.
               The result is an index set $I_k$ and an affine subspace $\bigcap_{i\in J_k} H_{\bar{k}_i,\bar{d}_i}$.
            b) Apply the membership algorithm to the $\ell_p$-body $B_n^{(p,V)}(t,\alpha)$
               and the affine subspace $H \cap \bigcap_{i\in J_k} H_{\bar{k}_i,\bar{d}_i}$.
               Either, the algorithm outputs an integer vector or it outputs
               that $B_n^{(p,V)}(t,\alpha) \cap H \cap \bigcap_{i\in J_k} H_{\bar{k}_i,\bar{d}_i}$ does not contain any integer
               vector.
        3. If there exists an index $k \in \mathbb{Z} \cap I_{\mathcal{B}\cap H}$ such that the algorithm out-
           puts an integer vector, output this vector. Otherwise, output that
           $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector.
```

Here, the last inequality follows since each eigenvalue of the matrix $V^T V$ is at most $n \cdot \text{size}(V^T V)$. Combining this with (6.3), this shows that every recursive instance consists of the original input general ℓ_p-ball $B_n^{(p,V)}(t, \alpha)$ together with an affine subspace $\tilde{H}$ of size of at most

$$\max\left\{\text{size}(H), r_{\mathcal{B}}^{n^{\mathcal{O}(1)}}\right\} = r^{n^{\mathcal{O}(1)}}. \tag{6.5}$$

In the following, we consider the size of each number computed by the algorithm in one recursion step. The input of such a recursion step consists of the general ℓ_p-ball $B_n^{(p,V)}(t, \alpha)$ together with an affine subspace $\tilde{H}$ of size of at most r^{n^c} for some fixed constant $c > 1$. This constant c depends only on the size of the general ℓ_p-ball.

If we apply the flatness algorithm with input of the general ℓ_p-ball $B_n^{(p,V)}(t, \alpha)$ and the affine subspace $\tilde{H}$, each number computed by the algorithm, particularly the vector $d_m \in \mathbb{Z}^n$ and the bounds of the interval $I_{\mathcal{B} \cap H}$ have size of at most

$$\left(r^{n^{\mathcal{O}(1)}}\right)^{p \cdot n^{\mathcal{O}(1)}} = r^{p \cdot n^{\mathcal{O}(1)}},$$

see Theorem 6.3.2. Also the other numbers computed by the membership algorithm in one recursion step have size of at most $r^{p \cdot n^{\mathcal{O}(1)}}$. Overall, we obtain that in every recursion step, the size of each number computed by the algorithm is at most $r^{p \cdot n^{\mathcal{O}(1)}}$.

Since the size of the affine subspace $\tilde{H}$ depends only on the size of the affine subspace H and the general ℓ_p-ball, this shows also that the size of each number computed by the lattice membership algorithm for general ℓ_p-balls has size of at most $r^{n^{\mathcal{O}(1)}}$. Moreover, this shows that the algorithm runs in polynomial space, see Proposition 6.1.5.

We now give an upper bound on the number of arithmetic operations of the algorithm. Let $T(m, n, p, r)$ be an upper bound on the number of arithmetic operations of the lattice membership algorithm for general ℓ_p-balls with input of a general ℓ_p-ball $B_n^{(p,V)}(t, \alpha)$ and an affine subspace H of dimension at most m, where r is an upper bound on the size of the general ℓ_p-ball and the affine subspace.

As we have seen in (6.5), each recursive instance consists of an affine subspace $\tilde{H}$ of size of at most $r^{n^{\mathcal{O}(1)}}$.

If the dimension of the affine subspace $\tilde{H}$ is 0, the vector $z \in \tilde{H}$ can be computed in $\mathcal{O}(n^3)$ arithmetic operations using Gaussian elimination and it can be checked if z is contained in $B_n^{(p,V)}(t, \alpha)$. Hence,

$$T(0, n, p, r_{\mathcal{B}}, r_H, r) = n^{\mathcal{O}(1)}.$$

If the dimension of the affine subspace $\tilde{H}$ is $k \geq 1$, the algorithm applies the flatness algorithm with input of the general ℓ_p-ball $B_n^{(p,V)}(t, \alpha)$ and the affine subspace $\tilde{H}$. Since the size of the affine subspace is at most $r^{n^{\mathcal{O}(1)}}$, the number of arithmetic operations of the flatness algorithm is at most

$$p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(k)} \cdot k^{k/(2e)} \leq p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(k)} k^{k/(2e)},$$

see Theorem 6.3.2. The number of arithmetic operations of the replacement procedure is polynomial in the dimension n and $\log_2(N)$. As we have seen above, see Inequality (6.4), N is at most $r^{n^{\mathcal{O}(1)}}$. Thus, the number of arithmetic operations of the replacement procedure is at most $(n \cdot \log_2(r))^{\mathcal{O}(1)}$.
Overall, this shows that in one recursion step the number of arithmetic operations of the lattice membership algorithm is at most

$$p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(k)} k^{k/(2e)},$$

where k is the dimension of the affine subspace defining the recursive instance.

If we consider a recursive instance where the dimension of the affine subspace is k, the number of recursive calls of the algorithm is determined by the length of the interval $I_{\mathcal{B} \cap \tilde{H}}$, which is at most $4k^2$, see Theorem 6.3.2. Overall, there exist at most

$$4m^2 \cdot 4(m-1)^2 \cdot \ldots \cdot 4(m-k+1)^2 = 2^{2(m-k)} \left(\frac{m!}{k!}\right)^2$$

different recursive instances, where the corresponding affine subspace has dimension k. This shows that the overall number of arithmetic operations of the lattice membership algorithm can be upper bounded by

$$T(m,n,p,r) \leq \sum_{k=0}^{m} p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(k)} k^{k/(2e)} 2^{2(m-k)} \left(\frac{m!}{k!}\right)^2.$$

Equivalently, there exist constants $c_1, c_2 > 1$ such that

$$\begin{aligned} T(m,n,p,r) &\leq \sum_{k=0}^{m} p(n \cdot \log_2(r))^{c_1} 2^{c_2 k} k^{k/(2e)} 2^{2m-k} \left(\frac{m!}{k!}\right)^2 \\ &\leq p(n \cdot \log_2(r))^{c_1} 2^{(c_2+2)m} \sum_{k=0}^{m} k^{k/(2e)} \left(\frac{m!}{k!}\right)^2. \end{aligned}$$

Using Stirling's formula, we see that $m! \leq m^m$ and that $k! \geq (k/e)^k$, see Section A.0.3 in the appendix. Thus, the number of arithmetic operations of the lattice membership algorithm can be upper bounded by

$$\begin{aligned} T(m,n,s,r) &\leq p(n \cdot \log_2(r))^{c_1} 2^{(c_2+2)m} \sum_{k=0}^{m} k^{k/(2e)} m^{2m} k^{-2k} e^{2k} \\ &\leq p(n \cdot \log_2(r))^{c_1} 2^{(c_2+2)m} m^{2m} e^{2m} \sum_{k=0}^{m} k^{(1/(2e)-2)k} \\ &\leq p(n \cdot \log_2(r))^{c_1} 2^{(c_2+2)m} m^{2m} e^{2m} m \\ &= p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} m^{2m}. \end{aligned}$$

□

If we apply the lattice membership algorithm with input of a general ℓ_p-ball and the whole vector space as subspace, the lattice membership algorithm solves the lattice membership problem.

Corollary 6.3.5. *The lattice membership algorithm for general ℓ_p-balls, Algorithm 10, solves the lattice membership problem for all general ℓ_p-balls $B_n^{(p,V)}(t,\alpha)$ correctly. The number of arithmetic operations is at most $p \cdot \log_2(r)^{\mathcal{O}(1)} n^{(2+o(1))n}$, where r is an upper bound on the size of the ℓ_p-ball. The algorithm runs in polynomial space and each number computed by the algorithm has size of at most $r^{p \cdot n^{\mathcal{O}(1)}}$, that means bit size of at most $p \cdot n^{\mathcal{O}(1)} \log_2(r)$.*

To complete the description of the lattice membership algorithm we need to describe a flatness algorithm for polytopes and general ℓ_p-balls. Furthermore, we need to present the replacement procedure. This will be done in the rest of this chapter. We start with the description of the flatness algorithms.

6.4. An algorithm for computing a flatness direction

In this section, we show that for polytopes and general ℓ_p-balls there exist flatness algorithms. The flatness algorithms are constructive versions of so-called flatness theorems.

We describe the flatness algorithm as a general algorithmic framework which works for classes of bounded convex sets. Given a bounded convex set $\mathcal{C}$ from such a class together with an affine subspace H of dimension m, the algorithm computes an $f(m)$-flatness direction of the convex set $\mathcal{C} \cap H$, i.e., a vector $d_m \in \mathbb{Z}^n \backslash \{0\}$ together with an interval $I_{\mathcal{C} \cap H}$ of length of at most $f(m)$ for some non-decreasing function $f : \mathbb{N} \to \mathbb{R}^{\geq 0}$. To recall, an $f(m)$-flatness direction of the set $\mathcal{C} \cap H$ is a vector $d_m \in \mathbb{Z}^n \backslash \{0\}$ such that there exists an interval $I_{\mathcal{C} \cap H}$ of length of at most $f(m)$ and the set $\mathcal{C} \cap H$ contains an integer vector if and only if there exists a hyperplane $H_{k,\tilde{d}}$, $k \in \mathbb{Z} \cap I_{\mathcal{C}}$, such that $\mathcal{C} \cap H \cap H_{k,\tilde{d}}$ contains an integer vector, see Definition 6.1.2. The interval is given through its upper and lower bound.

This section is organized as follows: We start with a general description of a flatness algorithm for bounded convex sets: First of all, we show that we can restrict ourselves to full-dimensional bounded convex sets. Then we show how we can realize a flatness algorithm for full-dimensional bounded convex sets. Here we start with special convex sets and later generalize this result to general convex sets by approximating the convex set with an approximate Löwner-John ellipsoid. Combining all this, we obtain a general description of a flatness algorithm for bounded convex sets. Finally, we will adapt this general framework to present concrete flatness algorithms for polytopes and general ℓ_p-balls.

6.4.1. A flatness algorithm for bounded convex sets

In the description of a general flatness algorithm, we consider bounded convex sets from some unspecified class $\mathcal{K}$. First of all, we show that we can restrict ourselves to full-dimensional bounded convex sets. We assume that we are given a full-dimensional bounded convex set $\mathcal{C}$ together with an affine subspace H which is given by a set of affine hyperplanes H_{k_i,d_i}, $m+1 \leq i \leq n$ for some parameter $m \leq n$.

Since the convex set $\mathcal{C} \cap H$ is not full-dimensional, we construct a bijective affine transformation which maps the convex set $\mathcal{C} \cap H$ to a convex set in $\mathbb{R}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$. Such a convex set can be identified with a full-dimensional convex set in $\mathbb{R}^m$. The important property of this transformation is that it is constructed in a way such that it maps every integer vector to an integer vector and vice versa. This guarantees that $\mathcal{C} \cap H$ contains an integer vector if and only if the corresponding convex set in $\mathbb{R}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$ contains an integer vector. The construction of such a transformation is described in the following.

First of all, we use an integer vector $v \in H$ to map the affine subspace H to the subspace $H - v$ which is given as the intersection of the affine hyperplanes H_{0,d_i}, $m+1 \leq i \leq n$. Since the normal vectors d_i of this subspace are linearly independent, they can be extended to a basis of the whole space $\mathbb{R}^n$, $B = [b_1, \ldots, b_m, d_{m+1}, \ldots, d_n]$. Obviously, every vector $x \in (H - v)$ satisfies $B^T x = (\bar{x}^T, 0^{n-m})^T$, where $\bar{x} \in \mathbb{R}^m$. That means, the function $x \mapsto B^T x$ maps the subspace $(H - v) = \bigcap_{i=m+1}^n H_{0,d_i}$ to the subspace $\bigcap_{i=m+1}^n H_{0,e_i}$. To guarantee that we obtain a bijection between the integer vectors in $H - v$ and $\bigcap_{i=m+1}^n H_{0,e_i}$, we construct a basis of the lattice $\mathcal{L}(B^T) \cap \bigcap_{i=m+1}^n H_{0,e_i}$ and map every vector in this lattice to its corresponding integer coefficient vector.

We observe that such a transformation can be constructed efficiently: Using the Hermite normal form, we can decide in polynomial time if there exists an integer vector in the affine subspace H and if so, compute one. This was shown by Frumkin, and von zur Gathen and Sieveking, see [Fru76b], [Fru76a], [vzGS76]. The basis $\bar{D}$ of the lattice $\mathcal{L}(B^T) \cap \bigcap_{i=m+1}^n H_{0,e_i}$ can be constructed efficiently by a polynomial time algorithm using the Smith normal form. Such an algorithm is presented by Micciancio in [Mic08].

Claim 6.4.1. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional bounded convex set. For $m \in \mathbb{N}$, $m < n$, let $H := \bigcap_{i=m+1}^n H_{k_i,d_i}$ be an affine subspace given by $d_i \in \mathbb{Z}^n$ linearly independent and $k_i \in \mathbb{Z}$. Let $v \in \mathbb{Z} \cap H$ and $B = [b_1, \ldots, b_m, d_{m+1}, \ldots, d_n] \in \mathbb{Z}^{n \times n}$ be a basis of $\mathbb{R}^n$ which contains the vectors d_i, $m+1 \leq i \leq n$. Let $\bar{D} \in \mathbb{Z}^{n \times m}$ be a basis of the lattice $\mathcal{L}(B^T) \cap \bigcap_{i=m+1}^n H_{0,e_i}$ and $\hat{D} := [\bar{D}, e_{m+1}, \ldots, e_n] \in \mathbb{Z}^{n \times n}$.*
Then, the bijective affine transformation

$$\tau : \mathbb{R}^n \to \mathbb{R}^n, \; x \mapsto \hat{D}^{-1} B^T (x - v)$$

satisfies the following properties:

- *The transformation τ is a bijective transformation between the affine subspace H and the subspace $\bigcap_{i=m+1}^n H_{0,e_i}$, $\tau(H) = \bigcap_{i=m+1}^n H_{0,e_i}$.*

- *The transformation τ is a bijective mapping between $\mathbb{Z}^n \cap H$ and $\mathbb{Z}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$.*

Proof. Obviously, the transformation τ is well-defined.
We start with the proof of the first statement. By definition of τ, for all $x \in \mathbb{R}^n$ and $m+1 \leq i \leq n$ we have that

$$\langle \tau(x), e_i \rangle = \langle \hat{D}^{-1} B^T (x-v), e_i \rangle = \langle B^T(x-v), (\hat{D}^T)^{-1} e_i \rangle. \tag{6.6}$$

Since the columns of $\bar{D}$ are vectors in $\mathbb{R}^n \cap \bigcap_{j=m+1}^n H_{0,e_j}$, we have $\bar{D}^T e_i = 0$ for all $m+1 \leq i \leq n$. Furthermore, $\hat{D}^T e_i = e_i$ for all $m+1 \leq i \leq n$. Combining this with (6.6), it follows that

$$\langle \tau(x), e_i \rangle = \langle B^T(x-v), e_i \rangle = \langle x-v, B \cdot e_i \rangle = \langle x-v, d_i \rangle.$$

Since $v \in H = \bigcap_{j=m+1}^n H_{k_j,d_j}$, we have

$$\langle \tau(x), e_i \rangle = \langle x, d_i \rangle - \langle v, d_i \rangle = k_i - k_i = 0$$

for all $m+1 \leq i \leq n$ and $x \in H$. This shows that $\tau(x) \in \bigcap_{j=m+1}^n H_{0,e_j}$. Since τ is bijective and the (affine) subspaces H and $\bigcap_{j=m+1}^n H_{0,e_j}$ have the same dimension, it follows that $\tau(H) = \bigcap_{j=m+1}^n H_{0,e_j}$. This proves the first statement.

We show the second statement in two steps. First, we show that τ maps every integer vector in H to an integer vector in $\mathbb{R}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$. Furthermore, we show that the inverse transformation τ^{-1} maps every integer vector in $\mathbb{R}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$ to an integer vector in H.

For every integer vector $x \in \mathbb{Z}^n$, we have $x - v \in \mathbb{Z}^n$ and $B^T(x-v) \in \mathcal{L}(B^T)$. As both x and v are contained in H, it follows that

$$\langle B^T(x-v), e_i \rangle = \langle x-v, Be_i \rangle = \langle x-v, d_i \rangle = 0$$

for all $m+1 \leq i \leq n$. This shows that $B^T(x-v)$ is a vector in the lattice $\mathcal{L}(B^T) \cap \bigcap_{j=m+1}^n H_{0,e_j}$. Since $\bar{D} \in \mathbb{Z}^{n \times m}$ is a basis of this lattice, there exists an integer vector $z \in \mathbb{Z}^m$ such that

$$\bar{D}z = B^T(x-v).$$

Obviously, the vector $z' = (z^T, 0^{n-m})^T \in \mathbb{Z}^n$ satisfies

$$\hat{D}z' = B^T(x-v).$$

From this, it follows that $\hat{D}^{-1}B^T(x-z) \in \mathbb{Z}^n$.

The inverse of the bijective affine transformation τ is given by

$$\tau^{-1} : \mathbb{R}^n \to \mathbb{R}^n, \; y \mapsto (B^T)^{-1}\hat{D}y + v.$$

To show that $\tau^{-1}(y) \in \mathbb{Z}^n$ for all integer vectors $y \in \mathbb{Z}^n \cap \bigcap_{j=m+1}^n H_{0,e_j}$, it is enough to show that $(B^T)^{-1}\hat{D}y \in \mathbb{Z}^n$.
Every integer vector $y' \in \mathbb{Z}^n \cap \bigcap_{j=m+1}^n H_{0,e_j}$ is of the form $y' = (y^T, 0^{n-m})^T$ with $y \in \mathbb{Z}^m$. Obviously, we have $\hat{D}y' = \bar{D}y$. Since $\bar{D}$ is a basis of the lattice $\mathcal{L}(B^T) \cap \bigcap_{j=m+1}^n H_{0,e_j}$, it follows that

$$\bar{D}y \in \mathcal{L}(B^T) \cap \bigcap_{j=m+1}^{n} H_{0,e_j} \subseteq \mathcal{L}(B^T).$$

Hence, there exists an integer vector $w \in \mathbb{Z}^n$ such that

$$\bar{D}y = B^T w.$$

□

We can now show that the transformation τ defined in Claim 6.4.1 can be used to obtain a γ-flatness direction of the convex set $\mathcal{C} \cap H$ from a γ-flatness direction of the full-dimensional convex set $\tau(\mathcal{C} \cap H)$.

Lemma 6.4.2. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional bounded convex set. For $m \in \mathbb{N}$, $m < n$, let $H := \bigcap_{i=m+1}^n H_{k_i,d_i}$ be an affine subspace given by $d_i \in \mathbb{Z}^n$ linearly independent and $k_i \in \mathbb{Z}$. Let $\tau : \mathbb{R}^n \to \mathbb{R}^n$ be the bijective affine transformation defined as in Claim 6.4.1. If $\tilde{d} \in \mathbb{Z}^m \backslash \{0\}$ is a γ-flatness direction of $\tau(\mathcal{C} \cap H)$ for some parameter $\gamma > 0$, then $d_m := (\hat{D}^{-1}B^T)^T(\tilde{d}^T, 0^{n-m})^T \in \mathbb{Z}^n \backslash \{0\}$ is a γ-flatness direction of $\mathcal{C} \cap H$.*

Proof. To prove this statement, we show that for all $\tilde{d} \in \mathbb{Z}^m \backslash \{0\}$ and $k \in \mathbb{Z}$ the set $\tau(\mathcal{C} \cap H) \cap H_{k,\tilde{d}}$ contains an integer vector if and only if the set $\mathcal{C} \cap H \cap H_{k+\langle v,d_m \rangle,d_m}$ contains an integer vector.

Obviously, for all $k \in \mathbb{R}$, the statement that $\tau(\mathcal{C} \cap H) \cap H_{k,\tilde{d}}$ contains an integer vector is equivalent to the statement that $\tau(\mathcal{C} \cap H) \cap H_{k,(\tilde{d}^T,0^{n-m})^T}$ contains an integer vector from $\mathbb{Z}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$ if we interpret $\tau(\mathcal{C} \cap H)$ as a convex set in $\mathbb{R}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$.

Since τ is a bijective mapping between $\mathbb{Z}^n \cap H$ and $\mathbb{Z}^n \cap \bigcap_{i=m+1}^n H_{0,e_i}$, the set $\tau(\mathcal{C} \cap H) \cap H_{k,(\tilde{d}^T,0^{n-m})^T}$ contains an integer vector if and only if $\mathcal{C} \cap H \cap \tau^{-1}(H_{k,(\tilde{d}^T,0^{n-m})^T})$ contains an integer vector. Since

$$\tau^{-1}\left(H_{k,(\tilde{d}^T,0^{n-m})^T}\right) = H_{k+\langle v,d_m \rangle,d_m},$$

it follows that $\tau(\mathcal{C} \cap H) \cap H_{k,\tilde{d}}$ contains an integer vector if and only if $\mathcal{C} \cap H \cap H_{k+\langle v,d_m \rangle,d_m}$ contains an integer vector. □

This result shows how we obtain an $f(m)$-flatness direction of the set $\mathcal{C} \cap H$, where $\mathcal{C}$ is a full-dimensional convex set from some class $\mathcal{K}$ and H is an affine subspace of dimension m. Still, we have to keep in mind that the class $\mathcal{K}$ needs to be closed under bijective affine transformation and intersection with hyperplanes. In the following, we consider

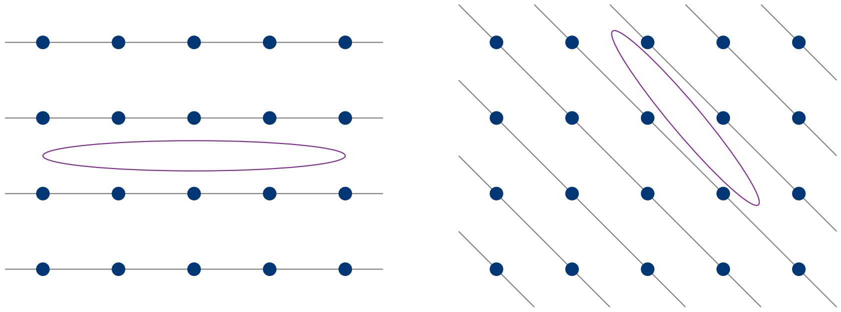

Figure 6.3.: **Idea behind the flatness theorems.** Full-dimensional convex sets which do not contain an integer vector are squeezed between parallel affine hyperplanes $H_{k,d}$, where $k \in \mathbb{Z}$, $d \in \mathbb{Z}^n \backslash \{0\}$.

such a class $\mathcal{K}$.

First of all, we show that we are able to realize a flatness algorithm for all full-dimensional bounded convex sets in this class $\mathcal{K}$. This flatness algorithm is an algorithmic realization of so-called flatness theorems. The basis of the flatness theorems is the following observation: If a full-dimensional convex set does not contain an integer vector, then it lies squeezed between the integer vectors. In other words: There exists a direction in which the convex set is flat. This observation is illustrated in Figure 6.3. Formally, this means that there exists a vector $\tilde{d} \in \mathbb{Z}^n$ such that only a bounded number of affine hyperplanes $H_{k,\tilde{d}}$, $k \in \mathbb{Z}$ intersect $\mathcal{C}$. The first result in this area was due to Khinchin, see [Khi48]. For an overview about the existing variants see [Bar02].

To formalize the idea how many hyperplanes intersect a bounded convex set, we use the notion of the width of a convex set $\mathcal{C} \subseteq \mathbb{R}^n$ along a vector $\tilde{d} \in \mathbb{R}^n \backslash \{0\}$, which is defined as the difference between the supremum and the infimum of the objective function $\langle \tilde{d}, x \rangle$, where $x \in \mathcal{C}$. Then, the width of $\mathcal{C}$ is defined as the minimal value of the width of $\mathcal{C}$ along a vector $\tilde{d}$, where $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$. A vector $\tilde{d}$ which achieves this minimum is called a flatness direction of $\mathcal{C}$.

Definition 6.4.3. *(Width of a convex set, flatness direction)*
Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set. For a vector $\tilde{d} \in \mathbb{R}^n \backslash \{0\}$ the width of $\mathcal{C}$ along $\tilde{d}$ *is defined as the number*

$$w_{\tilde{d}}(\mathcal{C}) := \sup\{\langle \tilde{d}, x \rangle | x \in \mathcal{C}\} - \inf\{\langle \tilde{d}, x \rangle | x \in \mathcal{C}\}.$$

The width of $\mathcal{C}$ *is defined as*

$$w(\mathcal{C}) := \min\{w_{\tilde{d}}(\mathcal{C}) | \tilde{d} \in \mathbb{Z}^n \backslash \{0\}\}.$$

A vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ with $w_{\tilde{d}}(\mathcal{C}) = w(\mathcal{C})$ is called a flatness direction of $\mathcal{C}$.

If the convex set $\mathcal{C}$ is closed, the objective function $\langle \tilde{d}, x\rangle$ achieves its extrema over $\mathcal{C}$, i.e., $w_{\tilde{d}}(\mathcal{C}) := \max\{\langle \tilde{d}, x\rangle | x \in \mathcal{C}\} - \min\{\langle \tilde{d}, x\rangle | x \in \mathcal{C}\}$. Using this notation, we can formulate the flatness theorems as follows: The width of every full-dimensional bounded convex body, which does not contain an integer vector, is less than a number which depends only on the dimension.

For certain classes of convex sets we are able to compute its width, for example for ellipsoids. Given an ellipsoid $E \subseteq \mathbb{R}^n$ we are able to compute its width together with a flatness direction since we are able to compute for a given vector $\tilde{d} \in \mathbb{R}^n$ the maximal and minimal value of $\langle \tilde{d}, x\rangle$, where $x \in E$.

Lemma 6.4.4. *Let $E = E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid and $d \in \mathbb{R}^n \backslash \{0\}$. Then*

$$\max\{\langle d, x\rangle | x \in E\} = \langle c, x\rangle + \sqrt{d^T D d} \text{ and}$$
$$\min\{\langle d, x\rangle | x \in E\} = \langle c, x\rangle - \sqrt{d^T D d}.$$

For $r > 0$ we have

$$\max\{\langle d, x\rangle | x \in r \star E\} = \langle c, x\rangle + r \cdot \sqrt{d^T D d} \text{ and}$$
$$\min\{\langle d, x\rangle | x \in r \star E\} = \langle c, x\rangle - r \cdot \sqrt{d^T D d}.$$

As a consequence, the width of an ellipsoid along d is

$$w_d(E) = 2\sqrt{d^T D d} \text{ and } w_d(r \star E) = 2r\sqrt{d^T D d}.$$

Proof. We start with the proof of the corresponding results for the Euclidean unit ball $\bar{B}_n^{(2)}(0,1)$. Using the Cauchy-Schwarz-inequality, we get that the value of the objective function $\langle d, x\rangle$, where $x \in \bar{B}_n^{(2)}(0,1)$, is at most

$$\max\left\{\langle d, x\rangle | x \in \bar{B}_n^{(2)}(0,1)\right\} \leq \max\left\{\|d\|_2 \cdot \|x\|_2 | x \in \bar{B}_n^{(2)}(0,1)\right\} \leq \|d\|_2.$$

If $x = d/\|d\|_2$, the Cauchy-Schwarz-inequality is fulfilled with equality, since

$$\langle d, x\rangle = \frac{\langle d, d\rangle}{\|d\|_2} = \|d\|_2.$$

Hence, $\max\{\langle d, x\rangle | x \in \bar{B}_n^{(2)}(0,1)\} = \|d\|_2$. With the same argumentation, we see that $\min\{\langle d, x\rangle | x \in B_n^{(2)}(0,1)\} = -\|d\|_2$.

The corresponding bounds for ellipsoids follow by straightforward computation: If $D = Q^T \cdot Q$ is a decomposition of the matrix D defining the ellipsoid, then the ellipsoid E is the image of the unit ball under the transformation $x \mapsto Q^T x + c$, see Lemma 2.2.7. This shows that

$$\begin{aligned}\{\langle d, x\rangle | x \in E\} &= \left\{\langle d, Q^T y + c\rangle \mid y \in \bar{B}_n^{(2)}(0,1)\right\} \\ &= \langle d, c\rangle + \left\{\langle Qd, y\rangle \mid y \in \bar{B}_n^{(2)}(0,1)\right\}.\end{aligned}$$

Now it follows directly from the observation above that the function $\langle d, x\rangle$ achieves its maximum/minimum over E if and only if the function $\langle Qd, y\rangle$ achieves is maximum/minimum over $\bar{B}_n^{(2)}(0,1)$. Hence,

$$\begin{aligned}\max\{\langle d, x\rangle | x \in E\} &= \langle d, c\rangle + \|Qd\|_2 \\ &= \langle d, c\rangle + \sqrt{d^T Q^T Q d} \\ &= \langle d, c\rangle + \sqrt{d^T D d}.\end{aligned}$$

The corresponding statement for the scaled ellipsoid $r \star E$ follows directly from the observation we made before together with the fact that $r \star E(D, c) = E(r^2 D, c)$, see Lemma 2.2.11,

$$\begin{aligned}\max\{\langle d, x\rangle | x \in E(r^2 D, c)\} &= \langle d, c\rangle + \sqrt{d^T r^2 D d} \\ &= \langle d, c\rangle + r \cdot \sqrt{d^T D d}.\end{aligned}$$

In the same way, we get the corresponding statements for the minimum. □

Now, we are able to show how a flatness direction of an ellipsoid can be computed. Additionally, we are able to show which hyperplanes of a family of hyperplanes have a non-empty intersection with an ellipsoid.

Proposition 6.4.5. *Let $E = E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid and $D = Q^T Q$ be an arbitrary decomposition of the matrix D. Then a vector $\tilde{d} \in \mathbb{Z}^n$ is a flatness direction of the ellipsoid if and only if $Q\tilde{d}$ is a shortest non-zero vector in the lattice $\mathcal{L}(Q)$. That means, we have*

$$w(E) = w_{\tilde{d}}(E) = 2\lambda_1^{(2)}(\mathcal{L}(Q))$$

and for $d = Q\tilde{d} \in \mathcal{L}(Q)$ we obtain

$$\begin{aligned}\max\{\langle \tilde{d}, x\rangle | x \in E\} &= \langle \tilde{d}, c\rangle + \|d\|_2 \text{ and} \\ \min\{\langle \tilde{d}, x\rangle | x \in E\} &= \langle \tilde{d}, c\rangle - \|d\|_2.\end{aligned}$$

Proof. As we have seen in Lemma 6.4.4, the width of an ellipsoid along a vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ is given by $w_{\tilde{d}}(E) = 2\sqrt{\tilde{d}^T D \tilde{d}}$. Hence, for every decomposition $D = Q^T Q$ of the matrix D, we have

$$\sqrt{\tilde{d}^T D \tilde{d}} = \sqrt{(Q\tilde{d})^T (Q\tilde{d})} = \|Q\tilde{d}\|_2 \tag{6.7}$$

which shows that the width $w_{\tilde{d}}(E)$ is minimized for $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ if $Q\tilde{d}$ is a shortest non-zero vector in the lattice $\mathcal{L}(Q)$ generated by the matrix Q. This proves the first statement. The proof of the other statements follows directly from (6.7). □

We observe that it follows from this proposition that the width of an ellipsoid can be computed using an arbitrary decomposition of the matrix defining the ellipsoid.

Remark 6.4.6. *The width of an ellipsoid $E(D,c)$ is independent from the chosen decomposition $D = Q^T \cdot Q$.*

With the results of Proposition 6.4.5, we are able to prove the flatness theorem for ellipsoids using the well-known transference bound for lattices due to Banaszczyk, see Theorem 3.3.6 in Chapter 3.

Theorem 6.4.7. *(Flatness Theorem for Ellipsoids)*
Let $E \subseteq \mathbb{R}^n$ be an ellipsoid. If the width of the ellipsoid is at least n, $w(E) \geq n$, then the ellipsoid contains an integer vector.

Proof. If the ellipsoid is given by the symmetric positive definite matrix $D \in \mathbb{R}^{n\times n}$ with $D = Q^TQ$ and the vector $c \in \mathbb{R}^n$, the ellipsoid E is the image of the Euclidean ball $\bar{B}_n^{(2)}(c,1)$ under the bijective linear transformation $x \mapsto Q^Tx$, see Lemma 2.2.7. Hence, the ellipsoid E does not contain an integer vector if and only if the Euclidean ball $\bar{B}_n^{(2)}((Q^T)^{-1}c,1)$ does not contain a vector from the lattice $\mathcal{L}((Q^T)^{-1})$. This shows that the covering radius of the lattice $\mathcal{L}((Q^T)^{-1})$ with respect to the Euclidean norm is greater than 1, since the distance from $(Q^T)^{-1}c$ to the lattice $\mathcal{L}((Q^T)^{-1})$ is greater than 1. Since $\mathcal{L}((Q^T)^{-1})^* = \mathcal{L}(Q)$, it follows from the transference bound, Theorem 3.3.6 in Chapter 3, that

$$\lambda_1^{(2)}(\mathcal{L}(Q)) < \mu^{(2)}(\mathcal{L}(\mathcal{L}(Q^T)^{-1})) \cdot \lambda_1^{(2)}(\mathcal{L}(Q)) \leq \frac{n}{2}.$$

Using that the width of E is exactly $2\lambda_1^{(2)}(\mathcal{L}(Q))$, this shows the statement. □

Proposition 6.4.5 together with the flatness theorem for ellipsoids provide a first idea of the realization of a flatness algorithm for ellipsoids: Given an ellipsoid, we compute its width and a corresponding flatness direction $\tilde{d} \in \mathbb{Z}^n$ by computing a shortest nonzero lattice vector. If the width is smaller than n, the interval $I_E = [\min\{\langle\tilde{d},x\rangle | x \in E\}, \max\{\langle\tilde{d},x\rangle | x \in E\}]$ has length at most n. This interval together with the flatness direction $\tilde{d}$ has the property that E contains an integer vector if and only if there exists an integer $k \in \mathbb{Z} \cap I$ such that $E \cap H_{k,\tilde{d}}$ contains an integer vector, i.e., $\tilde{d}$ is an n-flatness direction of E.
The flatness direction can be computed using Kannan's algorithm for SVP which we mentioned in Section 4.1 of Chapter 4, see Theorem 4.1.14. Even though this algorithm is not the fastest algorithm for SVP, it has the property that it runs in polynomial space, in contrast to the SVP-algorithm based on the computation of Voronoi cells from Micciancio and Voulgaris, see Theorem 4.1.13. A more formal description of the idea of a flatness algorithm for ellipsoids is presented in Algorithm 11.

The disadvantage of this approach is that it does not lead to a constructive algorithm: If the width of the ellipsoid E is larger than n, the flatness theorem for ellipsoids guarantees that E contains an integer vector, but we do not obtain an n-flatness direction.

Algorithm 11 Prototype of a flatness algorithm for ellipsoids

```
Input: Ellipsoid $E = E(D, c)$.
Used subroutine: Kannan's algorithm for SVP.
Output: A vector $\tilde{d} \in \mathbb{Z}^n$ together with an interval $I_E$ given by its upper and lower
  bound $k_{min}, k_{\max} \in \mathbb{Z}$.
  1. Compute a decomposition $D = Q^T Q$ of the matrix $D$.
  2. Compute a shortest non-zero lattice vector $d \in \mathcal{L}(Q)$.
     Let $\tilde{d} \leftarrow Q^{-1} d \in \mathbb{Z}^n$.
  3. Set $w \leftarrow 2\|d\|_2$.
     If $w \geq n$, output that $E$ contains an integer vector.
     Otherwise, output $\tilde{d} \in \mathbb{Z}^n$ together with
         $k_{\min} \leftarrow \langle \tilde{d}, c \rangle - \|d\|_2$ and
         $k_{\max} \leftarrow \langle \tilde{d}, c \rangle + \|d\|_2$.
```

To obtain a constructive algorithm, we use an idea of Dadush, Peikert, and Vempala, see [DPV11]. If the width w of the ellipsoid E along its flatness direction is strictly larger than the dimension n, we shrink the ellipsoid by the factor $n/w(E) < 1$, i.e., we consider the ellipsoid $E' := (n/w(E)) \star E$. This ellipsoid is completely contained in the original ellipsoid and its width is exactly n, see Lemma 6.4.4. Thus, if we find an integer vector in this ellipsoid, we have already found an integer vector in the original ellipsoid E. Since the width of the ellipsoid E' is exactly n, we obtain an interval $I_{E'}$ of length exactly n and there exists an integer $k \in \mathbb{Z} \cap I_{E'}$ such that $E' \cap H_{k,\tilde{d}}$ contains an integer vector. The complete algorithm is described in Algorithm 12.

In the next proposition, we show the correctness of the algorithm.

Proposition 6.4.8. *Given an ellipsoid $E \subseteq \mathbb{R}^n$, the flatness algorithm for ellipsoids, Algorithm 12, outputs an n-flatness direction $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ of the ellipsoid E together with a corresponding interval I_E of length at most n.*

Proof. The value w computed by the algorithm is the width of the ellipsoid E. The algorithm distinguishes between two cases:

- If $w \leq n$, the algorithm outputs the vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ together with the values

$$k_{\min} = \lceil \min\{\langle \tilde{d}, x \rangle | x \in E\} \rceil \text{ and}$$
$$k_{\max} = \lfloor \max\{\langle \tilde{d}, x \rangle | x \in E\} \rfloor,$$

  see Proposition 6.4.5. Since $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$, it holds in this case that E contains an integer vector if and only if there exists an integer $k \in \mathbb{Z}$ with $k_{\min} \leq k \leq k_{\max}$,

Algorithm 12 Flatness algorithm for ellipsoids

Input: Ellipsoid $E := E(D,c)$ with $D \in \mathbb{Q}^{n\times n}$ symmetric positive definite and $c \in \mathbb{Q}^n$.

Used subroutine: Kannan's algorithm for Svp.

Output: A vector $\tilde{d} \in \mathbb{Z}^n$ together with an interval I_E given by its upper and lower bound $k_{\min}, k_{\max} \in \mathbb{Z}$.

```
1. Compute a decomposition $D = Q^T Q$ of the matrix $D$.
2. Compute a shortest non-zero lattice vector $d \in \mathcal{L}(Q)$.
   Let $\tilde{d} := Q^{-1} d \in \mathbb{Z}^n$.
3. Set $w := 2\|d\|_2$.
   If $w \le n$, output $\tilde{d} \in \mathbb{Z}^n$ together with
       $k_{\min} \leftarrow \lceil \langle \tilde{d}, c\rangle - \|d\|_2 \rceil$ and
       $k_{\max} \leftarrow \lfloor \langle \tilde{d}, c\rangle + \|d\|_2 \rfloor$.
   Otherwise, output $\tilde{d} \in \mathbb{Z}^n$ together with
       $k_{\min} \leftarrow \lceil \langle \tilde{d}, c\rangle - \frac{n}{2} \rceil$ and
       $k_{\max} \leftarrow \lfloor \langle \tilde{d}, c\rangle + \frac{n}{2} \rfloor$.
```

such that $E \cap H_{k,\tilde{d}}$ contains an integer vector. Obviously, $k_{\min}$ and $k_{\max}$ define an interval of length at most n.

- If the width of E is greater than n, the width of the scaled ellipsoid $(n/w) \star E$ is exactly n, see Lemma 6.4.4. Thus it follows from the flatness theorem, that $(n/w) \star E$ contains an integer vector, see Theorem 6.4.7. Since $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$, every integer vector in this ellipsoid is contained in one of the affine hyperplanes $H_{k,\tilde{d}}$, where $k \in \mathbb{Z}$ with

$$\left\lceil \min\left\{ \langle \tilde{d}, x\rangle \middle| \, x \in (n/w) \star E \right\} \right\rceil \le k \le \left\lfloor \max\left\{ \langle \tilde{d}, x\rangle \middle| \, x \in (n/w) \star E \right\} \right\rfloor .$$

We have

$$\begin{aligned} \min\{\langle \tilde{d}, x\rangle | x \in (n/w) \star E\} &= \langle \tilde{d}, c\rangle - \frac{n}{w} \cdot \sqrt{\tilde{d}^T D \tilde{d}} \\ &= \langle \tilde{d}, c\rangle - \frac{n}{w} \cdot \frac{w}{2} \\ &= \langle \tilde{d}, c\rangle - \frac{n}{2} \end{aligned}$$

and

$$\max\{\langle \tilde{d}, x\rangle | x \in \frac{n}{w} \star E\} = \langle \tilde{d}, c\rangle + \frac{n}{2}.$$

Combining this with the fact that $(n/w) \star E \subseteq E$, this shows that there exists an index $k \in \mathbb{Z}$ with

$$\lceil \langle \tilde{d}, c \rangle - \frac{n}{2} \rceil \leq k \leq \lfloor \langle \tilde{d}, c \rangle + \frac{n}{2} \rfloor$$

such that $E \cap H_{k,\tilde{d}}$ contains an integer vector.

□

This result shows that the flatness algorithm for ellipsoids really computes an n-flatness direction of a given ellipsoid. It remains to show that the algorithm is polynomially space bounded. For this, we need to give an upper bound on the length of the flatness direction of the ellipsoid.

Lemma 6.4.9. *Let $D \in \mathbb{Q}^{n \times n}$ be a symmetric positive definite matrix. Let $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ be the flatness direction of the ellipsoid defined by the matrix D. Then*

$$\|\tilde{d}\|_2 \leq n^{(n+2)/2} \cdot \operatorname{size}(D)^{(n+1)/2}.$$

Proof. To prove an upper bound on the length of the vector $\tilde{d}$, we observe that $\tilde{d} = Q^{-1}d$, where v is a shortest non-zero lattice vector in $\mathcal{L}(Q)$, as we have seen in Proposition 6.4.5. Since the spectral norm and the Euclidean norm are compatible, this yields to the upper bound

$$\|\tilde{d}\|_2 = \|Q^{-1}d\|_2 \leq \|Q^{-1}\|_2 \cdot \|d\|_2.$$

The length of the vector d is the same as the length of a shortest vector in the lattice $\mathcal{L}(D^{1/2})$,

$$\lambda_1^{(2)}(\mathcal{L}(Q)) = \lambda_1^{(2)}(D^{1/2}),$$

as we observed in Remark 6.4.6. This shows that the length of the vector $\tilde{d} \in \mathbb{Z}^n$ is at most

$$\|\tilde{d}\|_2 \leq \|Q^{-1}\| \cdot \lambda_1^{(2)}(D^{1/2}). \tag{6.8}$$

Using Minkowski's first theorem, the Euclidean minimum distance of the lattice $\mathcal{L}(D^{1/2})$ is at most

$$\lambda_1^{(2)}(D^{1/2}) \leq \sqrt{n} \det(D^{1/2})^{1/n} = \sqrt{n} \det(D)^{1/2n}, \tag{6.9}$$

see Corollary 3.2.5. We can now give an upper bound on the spectral norm of the matrix Q^{-1}. Since the decomposition of a symmetric positive definite matrix in $D = Q^TQ$ is unique up to multiplication with an orthogonal matrix, there exists an orthogonal matrix $O \in \mathbb{R}^{n \times n}$ such that $O \cdot Q = D^{1/2}$. From this, one can show that the matrices $Q^{-1} = D^{-1/2} \cdot O$ and $D^{-1/2}$ have the same spectral norm:

$$\|Q^{-1}\|_2 = \sqrt{\eta_n(O^TD^{-1}O)} = \sqrt{\eta_n(D^{-1})} = \sqrt{\eta_n((D^{-1/2})^TD^{-1/2})} = \|D^{-1/2}\|_2,$$

where η_n denotes the largest eigenvalue of the matrix. The spectral norm of the matrix $\|D^{-1/2}\|_2$ is given by the square root of the spectral norm of D^{-1},

$$\|D^{-1/2}\|_2 = \sqrt{\eta_n(D^{-1})} = \|D^{-1}\|_2^{1/2},$$

where the spectral norm of D^{-1} is the inverse of an eigenvalue of D. It is easy to see, that each eigenvalue of the symmetric positive definite matrix is at least $1/\operatorname{size}(D)$, see for example [Ye92]. Hence, we obtain that

$$\|D^{-1}\|_2^{1/2} \leq \operatorname{size}(D)^{1/2}.$$

Combining this with (6.8) and (6.9), we obtain the following upper bound for the length of the vector $\tilde{d}$,

$$\|\tilde{d}\|_2 \leq \sqrt{n}\det(D)^{\frac{1}{2}(1+\frac{1}{n})}.$$

We have seen in Claim 2.2.18 in Chapter 2 that the determinant of a matrix D can be bounded by $\det(D) \leq (n \cdot \operatorname{size}(D))^n$. Hence, the length of the vector $\tilde{d}$ is at most

$$\|\tilde{d}\|_2 \leq \sqrt{n}(n \cdot \operatorname{size}(D))^{\frac{n}{2}(1+\frac{1}{n})} = \sqrt{n}(n \cdot \operatorname{size}(D))^{(n+1)/2}.$$

□

Now we can give an upper bound on the size of each number computed by the flatness algorithm. Furthermore, we give an upper bound on the number of arithmetic operations of the flatness algorithm for ellipsoids in the next proposition.

Proposition 6.4.10. *Given an ellipsoid $E = E(D, c) \subseteq \mathbb{R}^n$, where $D \in \mathbb{Q}^{n \times n}$ symmetric positive definite and $c \in \mathbb{Q}^n$, the number of arithmetic operations of the flatness algorithm for ellipsoids, Algorithm 12, is $2^{\mathcal{O}(n)}n^{n/(2e)}$. The algorithm is polynomially space bounded and each number computed by the algorithm has size of at most $r^{n^{\mathcal{O}(1)}}$, where r is an upper bound on the size of E and e is Euler's constant.*

Proof. Obviously, the algorithm runs in polynomial space if we can show that the size of each number computed by the flatness algorithm is at most polynomial in the size of the ellipsoid. This follows since Kannan's algorithm for Svp runs in polynomial space.

We have seen in Lemma 6.4.9 that the length of the flatness direction is at most

$$\|\tilde{d}\|_2 \leq n^{(n+2)/2}\operatorname{size}(D)^{(n+1)/2} \leq n^{(n+2)/2}r^{(n+1)/2}. \tag{6.10}$$

Since $\tilde{d}$ is an integer vector, this shows that $\operatorname{size}(\tilde{d}) \leq n^{(n+2)/2}r^{(n+1)/2}$. Hence, the only thing we need to take care of is that the numbers $k_{\min}, k_{\max} \in \mathbb{Z}$ are not getting too large. By definition, they are at most

$$\langle \tilde{d}, c\rangle + \min\left\{\frac{n}{2}, \|d\|_2\right\} \leq \langle \tilde{d}, c\rangle + \frac{n}{2}.$$

Combining the Cauchy-Schwarz inequality with (6.10), we obtain that

$$k \leq \|\tilde{d}\|_2 \cdot \|c\|_2 + \frac{n}{2} \leq n^{(n+2)/2} r^{(n+1)/2} \|c\|_2 + \frac{n}{2}.$$

Since r is an upper bound on the size of the ellipsoid E with center $c \in \mathbb{Q}^n$, we have $\|c\|_2 \leq \|c\|_1 \leq n \cdot r$ and we obtain that $k \leq r^{n^{\mathcal{O}(1)}}$.

The number of arithmetic operations is dominated by the number of arithmetic operations needed to compute a shortest non-zero lattice vector in $\mathcal{L}(Q)$ using Kannan's algorithm for SVP, see Theorem 4.1.14 in Chapter 4. This is at most $2^{\mathcal{O}(n)} n^{n/(2e)}$. □

To generalize this result to arbitrary bounded convex sets, we approximate the convex set by an approximate Löwner-John ellipsoid. To recall, for $0 < \gamma < 1/n$, a $(1/\gamma)$-approximate Löwner-John ellipsoid of an n-dimensional bounded convex set is an ellipsoid E with $E \subseteq \mathcal{C} \subseteq (1/\gamma) \star E$, see Definition 2.2.12 in Chapter 2.

Obviously, if we are able to compute approximate Löwner-John ellipsoids for a class of full-dimensional bounded convex sets, there exists a flatness algorithm for this class: Given an approximate Löwner-John ellipsoid E of a full-dimensional bounded convex set $\mathcal{C}$, we can compute the width and a corresponding flatness direction $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ of the ellipsoid. If this width is larger than n, the ellipsoid and therefore the convex set $\mathcal{C}$ contains an integer vector. In this case, in the same way as in the case of ellipsoids we obtain an interval $I_{\mathcal{C}}$ of length at most n, such that there exists an integer vector in $(n/w) \star E \cap H_{k,\tilde{d}} \subseteq \mathcal{C} \cap H_{k,\tilde{d}}$ for some integer $k \in \mathbb{Z} \cap I_{\mathcal{C}}$.
Otherwise, we observe that the width of the circumscribed ellipsoid $(1/\gamma) \star E$ is at most $(1/\gamma) \cdot w(E) \leq n/\gamma$ and that $\tilde{d} \in \mathbb{Z}^n$ is also a flatness direction of the circumscribed ellipsoid. Hence, the vector $\tilde{d} \in \mathbb{Z}^n$ satisfies that

$$\left| \left\lfloor \max\left\{ \langle \tilde{d}, x \rangle | x \in (1/\gamma) \star E \right\} \right\rfloor - \left\lceil \min\left\{ \langle \tilde{d}, x \rangle | x \in (1/\gamma) \star E \right\} \right\rceil \right| \leq \frac{n}{\gamma}.$$

Since the convex set $\mathcal{C}$ is contained in $(1/\gamma) \star E$, the vector $\tilde{d}$ also satisfies that every hyperplane $H_{k,\tilde{d}}$ which has a non-empty intersection with $\mathcal{C}$ satisfies

$$\min\left\{ \langle \tilde{d}, x \rangle | x \in (1/\gamma) \star E \right\} \leq k \leq \max\left\{ \langle \tilde{d}, x \rangle | x \in (1/\gamma) \star E \right\}.$$

Overall, this shows that for all full-dimensional bounded convex sets given together with a $(1/\gamma)$-approximate Löwner-John ellipsoid, we obtain a vector $\tilde{d} \in \mathbb{Z}^n \backslash \{0\}$ together with an interval $I_{\mathcal{C}}$ of length of at most n/γ such that the following holds: The convex set $\mathcal{C}$ contains an integer vector if and only if there exists a $k \in \mathbb{Z} \cap I_{\mathcal{C}}$ such that $\mathcal{C} \cap H_{k,\tilde{d}}$ contains an integer vector. Combining this with the observation made in Claim 6.4.1 we obtain a flatness algorithm for general bounded convex sets. A complete description of this approach is given in Algorithm 13.

Theorem 6.4.11. *Let $\mathcal{K}$ be a class of bounded convex sets closed under bijective affine transformation and intersection with hyperplanes orthogonal to the unit vectors and let*

Algorithm 13 Flatness algorithm for bounded convex sets

Input:

- A full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ from the class $\mathcal{K}$ which is closed under bijective affine transformation and intersection with hyperplanes orthogonal to the unit vectors and
- an affine subspace $H := \bigcap_{i=m+1}^{n} H_{k_i,d_i}$, where $d_i \in \mathbb{Z}^n$ linearly independent and $k_i \in \mathbb{Z}$ for all $m+1 \leq i \leq n$; alternatively $H = \mathbb{R}^n$.

Used subroutines:

- Rounding method for the class $\mathcal{K}$ which for a given full-dimensional convex set $\mathcal{C} \in \mathcal{K}$ computes a $(1/\gamma)$-approximate Löwner-John ellipsoid for some parameter $0 < \gamma \leq 1$ and
- Kannan's algorithm for SVP.

Output: A vector $d_m \in \mathbb{Z}^n \backslash \{0\}$ together with an interval $I_{\mathcal{C}}$ given by its upper and lower bound $k_{\min}, k_{\max} \in \mathbb{Z}$.

```
1. If $m = n$, set $v = 0$ and $\bar{V} = I_n$.
   Otherwise, compute $v \in \mathbb{Z} \cap H$, a basis $B = [b_1, \ldots, b_m, d_{m+1}, \ldots, d_n] \in \mathbb{Z}^{n \times n}$
         of $\mathbb{R}^n$. Compute a lattice basis $\bar{D} \in \mathbb{Z}^{n \times m}$ of $\mathcal{L}(B^T) \cap \bigcap_{i=m+1}^{n} H_{0,e_i}$.
         Set $\hat{D} := [\bar{D}, e_{m+1}, \ldots, e_n] \in \mathbb{Z}^n$ and $\bar{V} = \hat{D}^{-1} B^T$.
   Define the bijective mapping $\tau : \mathbb{R}^n \to \mathbb{R}^n$, $x \mapsto \bar{V}(x - v)$.

2. Apply the rounding method with input of the convex set $\tau(\mathcal{C} \cap H)$.
   The result is an ellipsoid $E(D, c) \subseteq \mathbb{R}^m$.

3. Compute a decomposition $D = Q^T Q$ of the matrix $D$.
   Compute a shortest non-zero lattice vector $d \in \mathcal{L}(Q)$.
   Let $\tilde{d} \leftarrow Q^{-1} d \in \mathbb{Z}^m$.

4. Set $w \leftarrow 2\|d\|_2$.
   If $w \leq m$, set
         $\tilde{k}_{\min} := \lceil \langle \tilde{d}, c \rangle - (1/\gamma) \cdot \|d\|_2 \rceil$ and
         $\tilde{k}_{\max} := \lfloor \langle \tilde{d}, c \rangle + (1/\gamma) \cdot \|d\|_2 \rfloor$.
   Otherwise, set
         $\tilde{k}_{\min} \leftarrow \lceil \langle \tilde{d}, c \rangle - m/2 \rceil$ and
         $\tilde{k}_{\max} \leftarrow \lfloor \langle \tilde{d}, c \rangle + m/2 \rfloor$.

5. Output the vector $d_m \leftarrow \bar{V}^T (\tilde{d}^T, 0^{n-m})^T$ together with
                    $k_{\min} \leftarrow \tilde{k}_{\min} + \langle v, d_m \rangle$ and
                    $k_{\max} \leftarrow \tilde{k}_{\max} + \langle v, d_m \rangle$.
```

$f : \mathbb{N} \to \mathbb{R}^{>0}$ be some non-decreasing function.
Assume that there exists a rounding method for this class, i.e., an algorithm which for a given full-dimensional convex set $\mathcal{C} \in \mathcal{K}$ computes a $(1/\gamma)$-approximate Löwner-John ellipsoid for some parameter $0 < \gamma \leq 1$.
Given a full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ from this class $\mathcal{K}$ and an affine subspace H of dimension m, the flatness algorithm for bounded convex sets, Algorithm 13, computes an (m/γ)-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $\mathcal{C} \cap H$ together with a corresponding interval $I_{\mathcal{C} \cap H}$ of length at most m/γ.

Proof. If the affine subspace H is not the whole space $\mathbb{R}^n$, the flatness algorithm computes the bijective affine transformation τ as described in Claim 6.4.1 which maps $\mathcal{C} \cap H$ to a full-dimensional bounded convex set in $\mathbb{R}^n$. Otherwise, we set τ as the identity. Since $\mathcal{K}$ is closed under bijective affine transformation and intersection with hyperplanes orthogonal to the unit vectors, we have $\tau(\mathcal{C} \cap H) \in \mathcal{K}$.

For this full-dimensional convex set $\tau(\mathcal{C} \cap H)$ in $\mathbb{R}^m$, the rounding method computes a $(1/\gamma)$-approximate Löwner-John ellipsoid E. For this ellipsoid, we compute its width w and a corresponding flatness direction $\tilde{d} \in \mathbb{Z}^m \backslash \{0\}$, see Lemma 6.4.5. Now, the algorithm distinguishes between two cases:

- If $w \leq m$, the algorithm computes an interval $[\tilde{k}_{\min}, \tilde{k}_{\max}]$, where
$$\tilde{k}_{\min} = \left\lceil \min \left\{ \langle \tilde{d}, x \rangle \mid x \in (1/\gamma) \star E(D, c) \right\} \right\rceil \text{ and}$$
$$\tilde{k}_{\max} = \left\lfloor \max \left\{ \langle \tilde{d}, x \rangle \mid x \in (1/\gamma) \star E(D, c) \right\} \right\rfloor,$$
see Lemma 6.4.4. Thus, this interval contains all integers $k \in \mathbb{Z}$ such that the affine hyperplane $H_{k,\tilde{d}}$ intersects $\tau(\mathcal{C} \cap H)$ and it follows that $\tau(\mathcal{C} \cap H)$ contains an integer vector if and only if there exists $k \in \mathbb{Z}$, $\tilde{k}_{\min} \leq k \leq \tilde{k}_{\max}$ such that $\tau(\mathcal{C} \cap H) \cap H_{k,\tilde{d}}$ contains an integer vector.
The length of the interval defined by $k_{\min}$ and $k_{\max}$ is at most the width of the ellipsoid $(1/\gamma) \star E$,
$$w((1/\gamma) \star E) = \max \left\{ \langle \tilde{d}, x \rangle \mid x \in (1/\gamma) \star E \right\} - \min \left\{ \langle \tilde{d}, x \rangle \mid x \in (1/\gamma) \star E \right\}.$$
Since the width of the ellipsoid $E(D, c)$ along the vector $\tilde{d}$ is at most m, the width of the ellipsoid $(1/\gamma) \star E(D, c)$ along $\tilde{d}$ is at most m/γ, i.e., $k_{\max} - k_{\min} \leq m/\gamma$.
- If $w > m$, we have $(m/w) \star E \subseteq E \subseteq \tau(\mathcal{C} \cap H)$.
The width of the ellipsoid $(m/w) \star E$ is exactly m. Hence, it is guaranteed by the flatness theorem that $(m/w) \star E \subseteq E \subseteq \tau(\mathcal{C} \cap H)$ contains an integer vector, see Theorem 6.4.7. This integer vector is contained in the intersection $(m/w) \star E \cap H_{k,\tilde{d}}$, where $k \in \mathbb{Z}$ satisfies
$$\left\lceil \min \left\{ \langle \tilde{d}, x \rangle \mid x \in (m/w) \star E \right\} \right\rceil \leq k \leq \left\lfloor \max \left\{ \langle \tilde{d}, x \rangle \mid x \in (m/w) \star E \right\} \right\rfloor.$$

Since

$$\min\left\{\langle\tilde{d},x\rangle \mid x\in(m/w)\star E\right\}=\langle\tilde{d},c\rangle-m/2 \text{ and}$$
$$\max\left\{\langle\tilde{d},x\rangle \mid x\in(m/w)\star E\right\}=\langle\tilde{d},c\rangle+m/2,$$

the interval defined by $\tilde{k}_{\min}$ and $\tilde{k}_{\max}$ guarantees that there exists an integer $k\in\mathbb{Z}$, $\tilde{k}_{\min}\le k\le\tilde{k}_{\max}$ such that $\tau(\mathcal{C}\cap H)\cap H_{k,\tilde{d}}$ contains an integer vector. Obviously, $\tilde{k}_{\max}-\tilde{k}_{\min}\le m$.

This shows that in both cases, the algorithm computes an (m/γ)-flatness direction $\tilde{d}\in\mathbb{Z}^m\backslash\{0\}$ of the convex set $\tau(\mathcal{C}\cap H)$ together with a corresponding interval $[\tilde{k}_{\min},\tilde{k}_{\max}]$. As we have seen in Lemma 6.4.2, this shows that the vector $d_m\in\mathbb{Z}^n\backslash\{0\}$ is an (m/γ)-flatness direction of $\mathcal{C}\cap H$. Additionally, we see that the numbers $k_{\min}$ and $k_{\max}$ defined as in the flatness algorithm define a corresponding interval. □

In Chapter 7, we will show that for polytopes and general ℓ_p-balls there exist deterministic algorithms that compute approximate Löwner-John ellipsoids. They are based on the famous ellipsoid method. At the moment, we use these results as black-boxes to obtain flatness algorithms for polytopes and general ℓ_p-balls.

6.4.2. A flatness algorithm for polytopes

Obviously, the class of polytopes is closed under bijective affine transformation and intersection with hyperplanes.

We will describe in Chapter 7 a rounding method for polytopes originally presented by Goffin in 1984. This rounding method is a polynomial time algorithm which computes for a full-dimensional polytope in $\mathbb{R}^n$ a $(1/\gamma)$-approximate Löwner-John ellipsoid, where $0<\gamma<1/n$. The proof of the following theorem together with a complete description of the algorithm appears in Chapter 7, Section 7.3.

Theorem 6.4.12. *Let $P=\{x\in\mathbb{R}^n|\langle a_i,x\rangle\le\beta_i$ for all $1\le i\le s\}$ with $a_i\in\mathbb{Z}^n,\beta_i\in\mathbb{Z}$ be a full-dimensional polytope. There exists a rounding method for polytopes that given such a polytope P together with a parameter γ with $0<\gamma<1/n$ computes a $1/\gamma$-approximate Löwner-John ellipsoid, i.e., a positive definite matrix $D\in\mathbb{Q}^{n\times n}$ and a vector $c\in\mathbb{Q}^n$ defining the ellipsoid $E(D,c)$ such that*

$$E(D,c)\subseteq P\subseteq\frac{1}{\gamma}\star E(D,c).$$

The number of arithmetic operations of the algorithm is

$$(ns\cdot\log_2(r))^{\mathcal{O}(1)},$$

where r is the size of the polytope. The algorithm runs in polynomial space and the size of the approximate Löwner-John ellipsoid is at most

$$2^{\mathcal{O}(n^4)}r^{\mathcal{O}(n)}.$$

Using this result, we can adapt the flatness algorithm for bounded convex sets and obtain a flatness algorithm for polytopes. A complete description of the algorithm is given in Algorithm 14.

Theorem 6.4.13. *(Theorem 6.2.1 restated.)*
Given a full-dimensional polytope $P \subseteq \mathbb{R}^n$ together with an affine subspace $H \subseteq \mathbb{R}^n$ of dimension m, the flatness algorithm for polytopes, Algorithm 14, computes a $2m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $P \cap H$ together with a corresponding interval $I_{P\cap H} \subseteq \mathbb{R}$ of length of at most $2m^2$. The number of arithmetic operations of the algorithm is

$$(ns \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} m^{m/(2e)},$$

where r is an upper bound on the size of the polytope, s is the number of constraints defining the polytope and e is Euler's constant. The algorithm runs in polynomial space and each number computed by the algorithm has size of at most $r^{n^{\mathcal{O}(1)}}$.

Proof. The transformation $\tau : x \mapsto \bar{V}(x-v)$ maps the intersection $P \cap H$ to the polytope $\{x \in \mathbb{R}^n | A\bar{V}^{-1}x \leq \beta - Av\} \cap \bigcap_{i=m+1}^{n} H_{0,e_i}$ which can be identified with the polytope $\{x \in \mathbb{R}^m | \tilde{A}x \leq \beta - Av\}$, where $\tilde{A} \in \mathbb{Z}^{s\times m}$ consists of the first m columns of the matrix $A\bar{V}^{-1}$. Combining this result with Theorem 6.4.11, it follows that the flatness algorithm for polytopes computes a $2m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $P \cap H$ together with a corresponding interval $I_{P\cap H}$ of length of at most $2m^2$.

Obviously, the flatness algorithm for polytopes is polynomially space bounded if each number computed by the algorithm has size of at most $r^{n^{\mathcal{O}(1)}}$, where r is an upper bound on the size of the polytope P and the affine subspace H.
The size of the polytope $\tilde{P}$ computed using the transformation τ in step 1 of the algorithm is of size of at most $r^{n^{\mathcal{O}(1)}}$. According to Theorem 6.4.12, the size of the approximate Löwner-John ellipsoid of the polytope $\tilde{P}$ computed by the rounding method is at most

$$2^{\mathcal{O}(n^4)}\,\mathrm{size}(\tilde{P})^{\mathcal{O}(n)} \leq 2^{\mathcal{O}(n^4)} r^{n^{\mathcal{O}(1)}}.$$

In fact, the flatness algorithm for polytopes combines the flatness algorithm for ellipsoids for the ellipsoid $E(D,c)$ and the ellipsoid $2m\star E(D,c)$. Hence, it follows from Proposition 6.4.10 that the size of each number computed by the algorithm is at most

$$\left(2^{\mathcal{O}(n^4)} r^{n^{\mathcal{O}(1)}}\right)^{n^{\mathcal{O}(1)}} = r^{n^{\mathcal{O}(1)}}.$$

Finally, we give an upper bound on the number of arithmetic operations of the flatness algorithm. Given a full-dimensional polytope in $\mathbb{R}^n$ together with an affine subspace of dimension $m > 0$, the computation of the affine bijective transformation in step 1 of the algorithm can be done using at most $n^{\mathcal{O}(1)}$ arithmetic operations. We apply the rounding method for polytopes with input of the polytope $\tilde{P}$ of size of at most $r^{n^{\mathcal{O}(1)}}$. Hence, it follows from Theorem 6.4.12 that the number of arithmetic operations of the rounding method is at most $(m \cdot s \log_2(r))^{\mathcal{O}(1)}$. For the computation of a shortest non-zero

Algorithm 14 Flatness algorithm for polytopes

```
Input:
  • A full-dimensional polytope $P \subseteq \mathbb{R}^n$ given by $A \in \mathbb{Z}^{s \times n}$ and $\beta \in \mathbb{Z}^s$ and
  • an affine subspace $H := \bigcap_{i=m+1}^{n} H_{k_i,d_i}$ where $k_i \in \mathbb{Z}^n$ linearly independent and
    $k_i \in \mathbb{Z}$ for all $m+1 \leq i \leq n$; alternatively $H := \mathbb{R}^n$.

Used subroutines: Kannan's algorithm for SVP, rounding method for polytopes.

Output: A vector $d_m \in \mathbb{Z}^n$ together with an interval $I_{P \cap H}$ given by its lower and upper
  bound $k_{\min}, k_{\max} \in \mathbb{Z}$.

  1. If $m = n$, set $v = 0$ and $\bar{V} = I_n$.
     Otherwise, compute $v \in \mathbb{Z} \cap H$, a basis $B = [b_1, \ldots, b_m, d_{m+1}, \ldots, d_n] \in \mathbb{Z}^{n \times n}$
           of $\mathbb{R}^n$.
           Compute a lattice basis $\bar{D} \in \mathbb{Z}^{n \times m}$ of $\mathcal{L}(B^T) \cap \bigcap_{i=m+1}^{n} H_{0,e_i}$.
           Set $\hat{D} \leftarrow [\bar{D}, e_{m+1}, \ldots, e_n] \in \mathbb{Z}^n$ and $\bar{V} \leftarrow \hat{D}^{-1} B^T$.
     Let $\tilde{P}$ be the polytope given by $\tilde{A} \in \mathbb{Z}^{s \times m}$ and $\beta - Av \in \mathbb{Z}^s$, where $\tilde{A}$ is the
     matrix which consists of the first $m$ columns of the matrix $A\bar{V}^{-1}$.

  2. Apply the rounding method for polytopes with input of the polytope $\tilde{P}$ and the
     parameter $\gamma = 1/(2m)$.
     The result is $D \in \mathbb{Q}^{m \times m}$ symmetric positive definite and $c \in \mathbb{Q}^m$.
     Compute a decomposition $D = Q^T Q$ of the matrix $D$.

  3. Compute a shortest non-zero lattice vector $d \in \mathcal{L}(Q)$.
     Let $\tilde{d} \leftarrow Q^{-1} d \in \mathbb{Z}^m$.

  4. Set $w := 2\|d\|_2$.
     If $w \leq m$, set
           $\tilde{k}_{\min} \leftarrow \lceil \langle \tilde{d}, c \rangle - 2m\|d\|_2 \rceil$ and
           $\tilde{k}_{\max} \leftarrow \lfloor \langle \tilde{d}, c \rangle + 2m\|d\|_2 \rfloor$.
     Otherwise, set
           $\tilde{k}_{\min} \leftarrow \lceil \langle \tilde{d}, c \rangle - m/2 \rceil$ and
           $\tilde{k}_{\max} \leftarrow \lfloor \langle \tilde{d}, c \rangle + m/2 \rfloor$.

  5. Output the vector $d_m \leftarrow \bar{V}^T (\tilde{d}^T, 0^{n-m})^T \in \mathbb{Z}^n$ together with
           $k_{\min} \leftarrow \tilde{k}_{\min} + \langle v, d_m \rangle$ and
           $k_{\max} \leftarrow \tilde{k}_{\max} + \langle v, d_m \rangle$.
```

vector using Kannan's algorithm we need at most $2^{\mathcal{O}(m)}m^{m/(2e)}$ arithmetic operations, see Theorem 4.1.14 in Chapter 4. This shows that the number of arithmetic operations of the rounding method is at most

$$n^{\mathcal{O}(1)} + (ns \cdot \log_2(r))^{\mathcal{O}(1)} + 2^{\mathcal{O}(m)}m^{m/(2e)} = (ns \cdot \log_2(r))^{\mathcal{O}(1)}\, 2^{\mathcal{O}(m)}m^{m/(2e)}.$$

□

6.4.3. A flatness algorithm for ℓ_p-bodies

Unfortunately, the class of general ℓ_p-balls as defined in Definition 6.3.1 is not closed under intersection with hyperplanes. Due to this reason, we consider a further generalization of ℓ_p-balls, the class of so-called ℓ_p-bodies.

For the construction of full-dimensional convex sets using the transformation defined in Claim 6.4.1, we need to consider the intersection of general ℓ_p-balls with hyperplanes orthogonal to the unit vectors, for example $B_n^{(p,V)}(t,\alpha) \cap H_{0,e_n}$. For simplicity, we denote this by $B_{n-1,n}^{(p,V)}(t,\alpha)$. To be precise, for $m \in \mathbb{N}$, $m \leq n$, we define

$$B_{m,n}^{(p,V)}(t,\alpha) := B_n^{(p,V)}(t,\alpha) \cap \bigcap_{i=m+1}^{n} H_{0,e_i}.$$

We will call these convex sets ℓ_p-*bodies*[1].

In the following, whenever we speak of an ℓ_p-body, we assume that we are given a nonsingular matrix $V \in \mathbb{R}^{n\times n}$, a vector $t \in \mathbb{R}^n$, parameters $m \in \mathbb{N}$, $m \leq n$, and $\alpha > 0$, and we consider the convex set $B_{m,n}^{(p,V)}(t,\alpha)$. The size of such an ℓ_p-body is the maximum of n, m, α and the size of the coordinates of V^{-1} and t.

We will interpret $B_{m,n}^{(p,V)}(t,\alpha)$ as a full-dimensional bounded convex set in the vector space $\mathbb{R}^m$. Then, we say that a vector $x \in \mathbb{R}^m$ is contained in $B_{m,n}^{(p,V)}(t,\alpha)$ if and only if $(x^T, 0^{n-m})^T \in B_n^{(p,V)}(t,\alpha)$.

To obtain a flatness algorithm for ℓ_p-bodies we need to be able to compute approximate Löwner-John ellipsoids for ℓ_p-bodies. In Chapter 7, we will present an algorithm which computes for a given ℓ_p-body an approximate Löwner-John ellipsoid with approximation factor $2/\gamma$ for $0 < \gamma < 1/n$. The algorithm is based on a variant of the ellipsoid method developed from Grötschel, Lovász and Schrijver in [GLS93]. The proof of the following theorem together with a complete description of the algorithm appears in Chapter 7, Section 7.2 of this thesis.

Theorem 6.4.14. *Let $B_{m,n}^{(p,V)}(t,\alpha) \subseteq \mathbb{R}^m$ be an ℓ_p-body given by $V \in \mathbb{Q}^{n\times n}$ nonsingular, $t \in \mathbb{Q}^n$, $\alpha > 0$ and $1 < p < \infty$. There exists a rounding method that given such a convex set together with a parameter γ with $0 < \gamma < 1/m$ satisfies the following properties:*

[1] Obviously, ℓ_p-bodies are not convex bodies but bounded convex sets.

- *Either it outputs that $B_{m,n}^{(p,V)}(t,\alpha)$ does not contain an integer vector, or*
- *it outputs a $2/\gamma$-approximate Löwner-John ellipsoid, i.e., a positive definite matrix $D \in \mathbb{Q}^{m\times m}$ and a vector $c \in \mathbb{Q}^m$ defining the ellipsoid $E(D,c)$ such that*

$$E(D,c) \subseteq B_{m,n}^{(p,V)}(t,\alpha) \subseteq \frac{2}{\gamma} \star E(D,c).$$

In this case, the size of the ellipsoid is at most $2^{\mathcal{O}(n^4)} r^{\mathcal{O}(n^2 p)}$.

The algorithm runs in polynomial space and its number of arithmetic operations is at most

$$\frac{p}{(1-m\gamma)^2}(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)}.$$

Here, r is an upper bound on the size of the ℓ_p-body.

Using this result, we obtain a flatness algorithm for ℓ_p-bodies in the same way as we obtain the flatness algorithm for polytopes. For a detailed description of the algorithm see Algorithm 15.

Theorem 6.4.15. *(Theorem 6.3.2 restated.)*
Given a general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ with $1 < p < \infty$ together with an affine subspace H of dimension m, the flatness algorithm for ℓ_p-bodies, Algorithm 15, outputs one of the following:

- *Either it outputs that $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector or*
- *it outputs a $4m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $B_n^{(p,V)}(t,\alpha) \cap H$ together with a corresponding interval $I_{\mathcal{B}\cap H} \subseteq \mathbb{R}$ of length at most $4m^2$.*

The number of arithmetic operations of the algorithm is

$$p \cdot (n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} m^{m/(2e)},$$

where r is an upper bound on the size of the general ℓ_p-ball and e is Euler's constant. The algorithm runs in polynomial space and each number computed by the algorithm has size of at most $r^{pn^{\mathcal{O}(1)}}$.

Proof. The transformation $\tau : x \mapsto \bar{V}(x - v)$ constructed in step 1 of the algorithm maps the intersection $B_n^{(p,V)}(t,\alpha) \cap H$ to the ℓ_p-body $B_{m,n}^{(p,\bar{V}V)}(\bar{V}(t-v),\alpha)$. By construction, it is guaranteed that $B_n^{(p,V)}(t,\alpha) \cap H$ contains an integer vector if and only if $B_{m,n}^{(p,\bar{V}V)}(\bar{V}(t-v),\alpha)$ contains an integer vector, see Claim 6.4.1.
Hence, if we apply the rounding method to the ℓ_p-body $B_{m,n}^{(p,\bar{V}V)}(\bar{V}(t-v),\alpha)$ and it outputs that $B_{m,n}^{(p,\bar{V}V)}(\bar{V}(t-v),\alpha)$ does not contain an integer vector, the intersection $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector.

Algorithm 15 Flatness algorithm for ℓ_p-bodies

Input:

- An ℓ_p-body $B_n^{(p,V)}(t,\alpha)$, where $V \in \mathbb{Q}^{n\times n}$ nonsingular, $t \in \mathbb{Q}^n$, $\alpha > 0$, $1 < p < \infty$, and
- an affine subspace $H = \bigcap_{i=m+1}^{n} H_{k_i,d_i}$, where $d_i \in \mathbb{Z}^n$ linearly independent and $k_i \in \mathbb{Z}$ for all $m+1 \leq i \leq n$; alternatively $H = \mathbb{R}^n$.

Used subroutines: Kannan's algorithm for Svp, rounding method for ℓ_p-bodies.

Output: A vector $d_m \in \mathbb{Z}^n$ together with an interval $I_{\mathcal{B}}$ given by its lower and upper bound $k_{\min}, k_{\max} \in \mathbb{Z}$ or the statement that $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector.

```
1. If $m = 0$, set $v = 0$ and $\bar{V} = I_n$.
   Otherwise, compute $v \in \mathbb{Z} \cap H$, a basis $B = [b_1, \dots, b_m, d_{m+1}, \dots, d_n] \in \mathbb{Z}^{n\times n}$
           of $\mathbb{R}^n$. Compute a lattice basis $\bar{D} \in \mathbb{Z}^{n\times m}$ of $\mathcal{L}(B^T) \cap \bigcap_{i=m+1}^{n} H_{0,e_i}$.
           Set $\hat{D} := [\bar{D}, e_{m+1}, \dots, e_n] \in \mathbb{Z}^n$ and $\bar{V} = \hat{D}^{-1}B^T$.
2. Apply the rounding method with input of the $\ell_p$-body $B_{m,n}^{(p,\bar{V}V)}(\bar{V}(t-v),\alpha)$ and
   the parameter $\gamma = 1/(2m)$.
3. If it outputs that $B_{m,n}^{(p,\bar{V}V)}(\bar{V}(t-v),\alpha)$ does not contain an integer vector, then
           output that $B_n^{(p,V)}(t,\alpha) \cap H$ does not contain an integer vector.
   Otherwise, the result is $D \in \mathbb{Q}^{m\times m}$ symmetric positive definite and $c \in \mathbb{Q}^m$.
       a) Compute a decomposition $D = Q^TQ$ of the matrix $D$.
       b) Compute a shortest non-zero lattice vector $d \in \mathcal{L}(Q)$.
          Let $\tilde{d} := Q^{-1}d \in \mathbb{Z}^m$.
       c) Set $w := 2\|d\|_2$.
          If $w \leq m$, set
              $\tilde{k}_{\min} := \lceil \langle \tilde{d}, c\rangle - 4m\|d\|_2 \rceil$ and
              $\tilde{k}_{\max} := \lfloor \langle \tilde{d}, c\rangle + 4m\|d\|_2 \rfloor$.
          Otherwise, set
              $\tilde{k}_{\min} \leftarrow \lceil \langle \tilde{d}, c\rangle - m/2 \rceil$ and
              $\tilde{k}_{\max} \leftarrow \lfloor \langle \tilde{d}, c\rangle + m/2 \rfloor$.
       d) Output the vector $d_m \leftarrow \bar{V}^T(\tilde{d}^T, 0^{n-m})^T \in \mathbb{Z}^n$ together with
                  $k_{\min} \leftarrow \tilde{k}_{\min} + \langle v, d_m\rangle$ and
                  $k_{\max} \leftarrow \tilde{k}_{\max} + \langle v, d_m\rangle$.
```

Otherwise, it follows from Theorem 6.4.11 that the flatness algorithm for ℓ_p-bodies computes a $4m^2$-flatness direction $d_m \in \mathbb{Z}^n \backslash \{0\}$ of $B_n^{(p,V)}(t,\alpha) \cap H$ together with a corresponding interval $I_{\mathcal{B}\cap H}$ of length at most $4m^2$.

Obviously, the flatness algorithm for ℓ_p-bodies is polynomially space bounded, if each number computed by the algorithm has size at most $r^{n^{\mathcal{O}(1)}}$, where r is an upper bound on the size of the general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ and the affine subspace H. Especially, we use here that Kannan's algorithm for SVP is polynomially space bounded, see Theorem 4.1.14 in Chapter 4.
The bijective affine transformation constructed in step 1 of the algorithm is given by a matrix and a vector whose size is at most size$(H)^{n^{\mathcal{O}(1)}} = r^{n^{\mathcal{O}(1)}}$. It follows that the size of the ℓ_p-body $B_{m,n}^{(p,\bar{V}V)}(\bar{V}(t-v),\alpha)$ is at most $r^{n^{\mathcal{O}(1)}}$.
According to Theorem 6.4.14, the size of an approximate Löwner-John ellipsoid computed by the rounding method with $\gamma = 1/(2m)$ is at most $2^{\mathcal{O}(n^4)} r^{\mathcal{O}(n^2 p)}$. Since the flatness algorithm is a combination of the flatness algorithm for ellipsoids applied with the inscribed ellipsoid $E(D,c)$ and the circumscribed ellipsoid $4m \star E(D,c)$, it follows from Proposition 6.4.10 that the size of each number computed by the algorithm is at most

$$\left(2^{\mathcal{O}(n^4)} r^{\mathcal{O}(n^2 p)}\right)^{n^{\mathcal{O}(1)}} = r^{p \cdot n^{\mathcal{O}(1)}}.$$

Finally, we give an upper bound on the number of arithmetic operations of the flatness algorithm. Given a general ℓ_p-ball in $\mathbb{R}^n$ together with an affine subspace of dimension $m > 0$, the computation of the bijective affine transformation in step 1 of the algorithm can be done using at most $n^{\mathcal{O}(1)}$ arithmetic operations. We apply the rounding method for ℓ_p-bodies with input of an ℓ_p-body of size of at most $r^{n^{\mathcal{O}(1)}}$. Hence, it follows from Theorem 6.4.14 that the number of arithmetic operations of the rounding method is at most $p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)}$. The number of arithmetic operations of Kannan's algorithm for SVP is upper bounded by $2^{\mathcal{O}(m)} m^{m/(2e)}$, see Theorem 4.1.14 in Chapter 4. This shows that the number of arithmetic operations of the rounding method is upper bounded by

$$n^{\mathcal{O}(1)} + p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} + 2^{\mathcal{O}(m)} m^{m/(2e)} = p(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)} m^{m/(2e)}.$$

□

6.5. Replacement procedure

In our algorithm for the lattice membership problem we assumed that we have access to a so-called replacement procedure. The replacement procedure gets an affine subspace $H \subseteq \mathbb{R}^n$, an additional hyperplane $H_{k,d}$, and some parameter $N \in \mathbb{N}$ as input and computes a set of new hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$ for some index set J, with small size in polynomial time. For each integer vector which is contained in an ℓ_1-ball with radius less than N it can be guaranteed that it is contained in the affine subspace $H \cap H_{k,\tilde{d}}$ if

and only if it is contained in the intersection $H \cap \bigcap_{i \in J} H_{\bar{k}_i, \bar{d}_i}$. This means, that if the parameter N is chosen appropriately depending on the shape of some convex set $\mathcal{C}$, it can be guaranteed that each integer vector from $\mathcal{C}$ is contained in the affine subspace $H \cap H_{k,d}$ if and only if it is contained in the intersection $H \cap \bigcap_{i \in J} H_{\bar{k}_i, \bar{d}_i}$. Furthermore, if the affine subspace H and the affine hyperplane $H_{k,d}$ are affinely independent, then the affine subspace $\bigcap_{i \in J} H_{\bar{k}_i, \bar{d}_i}$ are affinely independent.
This replacement procedure was used in the lattice membership algorithm to make the algorithm run in polynomial space. In this section, we will describe this replacement procedure and we will prove Theorem 6.1.6.

Originally, the replacement procedure was developed by Frank and Tardos in 1987 as a preprocessing technique to make certain polynomial time algorithms for linear programming strongly polynomial time[2]. For an overview about this application of the replacement procedure see [Eis10].
Kannan observed that the preprocessing technique could also be used to make Lenstra's algorithm for integer programming [Len83] or its improvement by Kannan [Kan87b] run in polynomial space.
The replacement procedure described in the following is a slight generalization of the replacement procedure developed by Frank and Tardos adapted to our context. It can be used to make the lattice membership algorithm as we presented in Section 6.1 run in polynomial space.

The main idea of the replacement procedure as follows: For a given hyperplane $H_{k,d} \subseteq \mathbb{R}^n$, a vector $b \in \mathbb{R}^n$ is contained in the hyperplane $H_{k,d}$ if and only if the vector $(b^T, -1)^T \in \mathbb{R}^{n+1}$ is contained in the hyperplane $H_{0,w}$, where $w = (d^T, k)^T \in \mathbb{R}^{n+1}$. Now, we can show that there exists a decomposition procedure that computes a representation for the vector w as a linear combination of integer vectors with small size. Additionally, the coefficients of this representation build a rapidly decreasing sequence.

The decomposition procedure used to compute such a representation is a kind of multidimensional continued fraction expansion. The techniques used in this replacement procedure are completely independent from the techniques presented so far. The main tool is simultaneous Diophantine approximation.

Simultaneous Diophantine approximation

Simultaneous Diophantine approximation deals with the topic of considering n real numbers $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ in order to approximate them simultaneously by rational numbers. Here, simultaneously means, that the rational numbers have the same denominator. A fundamental result due to Dirichlet shows that for all $N \in \mathbb{N}$ there exists a simultaneous approximation for arbitrary numbers $\alpha_1, \ldots, \alpha_n$, where the common denominator q is at most N^n.

[2] Loosely speaking, a polynomial time algorithm is strongly polynomial time if the number of arithmetic operations depends not on the binary encoding length of the input.

Theorem 6.5.1. *(Dirichlet's Theorem about simultaneous approximation)*
Let $N \in \mathbb{N}$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. Then, there exists $p_1, \ldots, p_n \in \mathbb{Z}$ and $q \in \mathbb{Z}$ such that

$$1 \leq q \leq N^n \text{ and } |q \cdot \alpha_i - p_i| < \frac{1}{N} \text{ for all } 1 \leq i \leq n.$$

Dirichlet's Theorem guarantees the existence of a simultaneous approximation with approximation factor at most $q \cdot N^n \leq N^{n+1}$, since we have $|\alpha_i - p_i/q| < (q \cdot N)^{-1}$ for all $1 \leq i \leq n$. A proof of this result can for example be found in [Sch91].

We observe that for $N \geq 2$, the numbers p_i are uniquely characterized as the integers which are next to $q \cdot \alpha_i$, $1 \leq i \leq n$. Hence, if there exists an index i such that $\alpha_i = 0$, for all $q \in \mathbb{N}$ $p_i = 0$ is the only number which satisfies $|q \cdot \alpha_i - p_i| < 1/N \leq 1$.
As a consequence, if $\alpha_i \in \mathbb{Z}$, then $p_i = \alpha_i \cdot q \in \mathbb{Z}$ is the only number which satisfies $|q \cdot \alpha_i - p_i| < 1/N \leq 1$.

Dirichlet's result itself is not constructive but the LLL-algorithm from 1982, see Theorem 4.1.10 in Chapter 4, can be used to compute a simultaneous Diophantine approximation for given rational numbers in polynomial time. Since this application of the LLL-algorithm was observed by Lovász, the algorithm is named Lovász's approximation algorithm. The common denominator computed by the algorithm is at most $2^{n^2} N^{n+1}$.

Theorem 6.5.2. *(Lovász's approximation algorithm, [LLL82])*
There exists an algorithm, which computes for $N \in \mathbb{N}$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{Q}$ integers $p_1, \ldots, p_n \in \mathbb{Z}$ and $q \in \mathbb{N}$ such that

$$1 \leq q \leq 2^{n^2} N^n \text{ and } |q \cdot \alpha_i - p_i| < \frac{1}{N} \text{ for all } 1 \leq i \leq n.$$

The algorithm is called Lovász's approximation algorithm. The number of arithmetic operations of this algorithm is at most $n^6 \log_2(B)$, where B is an upper bound on the size of the values α_i, $1 \leq i \leq n$, and N. Each number computed by the algorithm has size at most $\mathcal{O}(B^{n^3})$.

A description of Lovász's approximation algorithm can be found in [LLL82] or for example in [vzGG03]. The disadvantage of this algorithm is that the number of arithmetic operations depends on the size of the input. Yet, if we restrict the input to rationals with absolute value of at most 1, we can construct an algorithm for simultaneous Diophantine approximation whose number of arithmetic operations depends only on the parameter $N \in \mathbb{N}$ and on the number n of rationals. Conversely, the upper bound on the common denominator q gets worse than the upper bound of the common denominator computed by Lovász's algorithm by the factor 2^n to $2^{n^2+n} N^{n+1}$.

The algorithm is called the revised simultaneous approximation algorithm. As input, it gets an integer $N \in \mathbb{N}$ and rationals $\alpha_1, \ldots, \alpha_n \in \mathbb{Q}$ from the interval between -1 and 1. In the first step, it computes an individual rational approximation α_i' for each rational number α_i whose size depends only on n and N, i.e., is independent of the size of α_i itself.

Algorithm 16 Revised simultaneous approximation algorithm

Input:

- A parameter $N \in \mathbb{N}$,
- numbers $\alpha_1, \dots, \alpha_n \in \mathbb{Q}$ satisfying $|\alpha_i| \leq 1$ for all $1 \leq i \leq n$.

Used subroutine: Lovász's approximation algorithm.

Output: Numbers $p_1, \dots, p_n \in \mathbb{Z}$, $q \in \mathbb{N}$

```
1. For $1 \leq i \leq n$, set
       $$\alpha'_i := -\frac{\lfloor \alpha_i 2^{n^2+n+1} N^{n+1} \rfloor}{2^{n^2+n+1} N^{n+1}} \text{ and } N' := 2N.$$
2. Apply Lovász's approximation algorithm with input of the parameter $N'$ and the
   numbers $\alpha'_1, \dots, \alpha'_n$. We obtain $p_1, \dots, p_n \in \mathbb{Z}$ and $q \in \mathbb{N}$.
3. Output $p_1, \dots, p_n$ and $q$.
```

Hence, if we apply Lovász's approximation algorithm to these rationals α'_i, $1 \leq i \leq n$, the number of arithmetic operations depends only on n and N. Using the triangle inequality, it can be shown that the rational approximations computed by Lovász's approximation algorithm are also good (enough) approximations for the input numbers α_i, $1 \leq i \leq n$. A detailed description of the algorithm is given in Algorithm 16.

Lemma 6.5.3. *Given $N \in \mathbb{N}$, $\alpha_1, \dots, \alpha_n \in \mathbb{Q}$ with $|\alpha_i| \leq 1$ for $1 \leq i \leq n$, the revised simultaneous approximation algorithm computes numbers $p_1, \dots, p_n \in \mathbb{Z}$ and $q \in \mathbb{N}$ such that*

$$1 \leq q \leq 2^{n^2+n} N^n \text{ and } |q\alpha_i - p_i| < \frac{1}{N} \text{ for all } 1 \leq i \leq n.$$

The number of arithmetic operations of the revised simultaneous approximation algorithm is polynomial in n and $\log_2(N)$, i.e., $(n \cdot \log_2(N))^{\mathcal{O}(1)}$. Each number computed by the algorithm has size of at most $\max\{2^{n^{\mathcal{O}(1)}} N^{\mathcal{O}(n)}, B\}$, where B is an upper bound on the size of the value α_i, $1 \leq i \leq n$, and N.

Proof. For each index i, $1 \leq i \leq n$, the number α'_i approximates the number α_i with a factor $2^{n^2+n+1} N^{n+1}$ since by definition they satisfy $|2^{n^2+n+1} N^{n+1}(\alpha_i - \alpha'_i)| \leq 1$. That means, the difference between α_i and α'_i is at most

$$|\alpha_i - \alpha'_i| \leq 2^{-(n^2+n+1)} N^{-(n+1)}. \tag{6.11}$$

Lovász's approximation algorithm with input of the numbers $\alpha'_1, \dots, \alpha'_n$ and $N' = 2N$ computes a number $q \in \mathbb{N}$ satisfying

$$q \leq 2^{n^2} N'^n = 2^{n^2+n} N^n. \tag{6.12}$$

Additionally, for each index i, $1 \leq i \leq n$, it computes a number $p_i \in \mathbb{N}$ that approximates α'_i with the factor $q \cdot N'$, i.e.,

$$|q \cdot \alpha'_i - p_i| < \frac{1}{N'}, \tag{6.13}$$

see Theorem 6.5.2. Using the triangle inequality, it follows that p_i also approximates α_i,

$$|q \cdot \alpha_i - p_i| \leq q \cdot |\alpha_i - \alpha'_i| + |q\alpha'_i - p_i|.$$

Combining (6.11), (6.12) and (6.13), we see that this is less than

$$2^{n^2+n} N^n \cdot 2^{-(n^2+n+1)} N^{-(n+1)} + (2N)^{-1} = N^{-1}$$

which shows that the revised simultaneous approximation algorithm computes an approximation with approximation factor $2^{n^2+n} N^{n+1}$.

The revised simultaneous approximation algorithm applies Lovász's approximation algorithm with input numbers of size of at most $2^{n^{\mathcal{O}(1)}} N^{\mathcal{O}(n)}$. Hence, the size of each number computed by the revised simultaneous approximation algorithm is upper bounded by $\max\{2^{n^{\mathcal{O}(1)}}, B\}$, where B is an upper bound on the size of the values α_i, $1 \leq i \leq n$, and N.

The number of arithmetic operations of Lovász's approximation algorithm is at most $n^6 \log_2(B)$, where B is an upper bound on the size of the numbers α'_i, $1 \leq i \leq n$, and N'. Since α_i is at most 1, the size of the numbers α'_i is at most $2^{n^2+n+1} N^{n+1}$. This shows that the number of arithmetic operations of Lovász's approximation algorithm applied in the revised approximation algorithm is at most

$$n^6 \log_2(2^{n^2+n+1} N^{n+1}) = n^{\mathcal{O}(1)} \log_2(N).$$

For the computation of the numbers α'_i, we need to compute the greatest integer smaller than $\alpha_i 2^{n^2+n+1} N^{n+1}$. This can be done using binary search. Since the absolute value of the numbers α_i is at most 1, the number of elements on which we perform the binary search is at most

$$\mathcal{O}(\log_2(2^{n^2+n+1} N^{n+1})) = \mathcal{O}(n^2 \log_2(N)).$$

Since we need to do this only for numbers whose absolute value is at most $2^{n^2+n+1} N^{n+1}$, the number of arithmetic operations to do this is at most polynomial in n and $\log_2(N)$. Hence, the number of arithmetic operations of the revised simultaneous approximation algorithm is polynomial in n and $\log_2(N)$. □

Decomposition algorithm

We now show that the revised simultaneous approximation algorithm can be used to represent an arbitrary vector $w \in \mathbb{Q}^n$ as a positive linear combination of at most n integer

Algorithm 17 Decomposition Algorithm

```
Input:
  • A parameter $N \in \mathbb{N}$ and
  • a vector $w \in \mathbb{Q}^n \backslash \{0\}$.
Used subroutine: Revised simultaneous approximation algorithm
Output: Integer vectors $v_1, \ldots, v_k \in \mathbb{Z}^n$ together with $\chi_1, \ldots, \chi_k \in \mathbb{Q}$
  1. Set $w_0 \leftarrow w$ and $k \leftarrow -1$.
  2. For $w_{k+1} \neq 0$,
     a) set $k \leftarrow k+1$ and $w'_k \leftarrow w_k / \|w_k\|_\infty$.
     b) Apply the revised simultaneous approximation algorithm with input of the
        parameter $N$ and the coordinates $w_k(1), \ldots, w_k(n)$ of the vector $w'_k$.
        We obtain $v_{k+1}(1), \ldots, v_{k+1}(n) \in \mathbb{Z}$ and $q_{k+1} \in \mathbb{N}$.
     c) Set
           $v_{k+1} \leftarrow (v_{k+1}(1), \ldots, v_{k+1}(n))^T$,
           $\chi_{k+1} \leftarrow \frac{\|w_k\|_\infty}{q_{k+1}}$, and
           $w_{k+1} \leftarrow w_k - \chi_{k+1} \cdot v_{k+1}$.
     d) Output $v_1, \ldots, v_{k+1} \in \mathbb{Z}^n$ and $\chi_1, \ldots, \chi_{k+1} \in \mathbb{Q}$.
```

vectors $v_1, \ldots, v_n \in \mathbb{Z}^n$, whose components are relatively small. Additionally, this representation has the property that the coefficients of this representation decrease very fast.

The idea of this algorithm is easy. By scaling a vector w with its largest coefficient $\|w\|_\infty$, we achieve a vector whose coefficients have absolute value at most 1. Hence, the revised simultaneous approximation algorithm can be used to compute a simultaneous approximation of these coefficients in form of integers $v_1(1), \ldots, v_1(n) \in \mathbb{Z}$ and a common denominator $q_1 \in \mathbb{N}$.
We set $v_1 \in \mathbb{Z}^n$ as the vector with the coefficients $v_1(i)$, $1 \leq i \leq n$. Now, we want to represent the vector $w - (\|w_0\|_\infty / q_1) v_1$ as a positive linear combination of integer vectors with small coefficients. This can be done recursively in the same way as for the vector w. The algorithm terminates if we obtain the vector 0. A detailed description of the algorithm is given in Algorithm 17.

Theorem 6.5.4. *The decomposition algorithm with input of the vector $w \in \mathbb{Q}^n$ and the parameter $N \in \mathbb{N}$, $N \geq 2$, computes vectors $v_1, \ldots, v_k \in \mathbb{Z}^n$, $k \leq n$, and numbers*

$\chi_1, \ldots, \chi_k > 0$ *such that the following holds:*

- *The vector* w *is a linear combination of the vectors* $v_1, \ldots v_k$ *with the coefficients* χ_i*, i.e., we have* $w = \sum_{i=1}^{k} \chi_i v_i$.
- *The size of the vectors* v_i *is at most* $2^{n^2+n}N^n$*, i.e.,*

$$\|v_i\|_\infty \leq 2^{n^2+n}N^n \text{ for all } 1 \leq i \leq k.$$

- *The components of this linear representation decrease, i.e. for* $2 \leq j \leq k$*,*

$$\|\sum_{i=j}^{n} \chi_i v_i\|_\infty < \frac{\chi_{j-1}}{N}.$$

  Especially, we have

$$\chi_j < \frac{1}{N\|v_j\|_\infty}\chi_{j-1}.$$

The number of arithmetic operations of the decomposition algorithm is polynomial in n *and* $\log_2(N)$*, i.e.,* $(n \cdot \log_2(N))^{\mathcal{O}(1)}$.

Proof. To show that the algorithm terminates after at most n steps of iteration, we show that in each iteration step the number of non-zero coordinates decreases: For all $i \geq 1$, the number of non-zero components of the vector w_i is strictly smaller than the number of non-zero components of the vector w_{i-1}.

- First, we show that every component $w_{i-1}(j)$ which is zero, remains zero, i.e., $w_i(j) = 0$. This follows since the corresponding approximation $v_i(j)$ computed by the revised simultaneous approximation algorithm is 0. Hence,

$$w_i(j) = w_{i-1}(j) - \chi_i \cdot v_i(j) = 0.$$

  This shows that the number of non-zero coordinates of the vector w_i is not greater than the number of non-zero coordinates of the vector w_{i-1}.

- Now, we show that the coordinates of w_{i-1} with maximal value become zero. Let $1 \leq j \leq n$ be an index with $|w_{i-1}(j)| = \|w_{i-1}\|_\infty$. Then, we have

$$w'_{i-1}(j) = \text{sign}(w_{i-1}(j)) \in \mathbb{Z}.$$

  Since the number $v_i(j)$ computed by the revised simultaneous approximation algorithm is the closest integer to $q_i \cdot w'_{i-1}(j)$, the corresponding approximation $v_i(j)$ is $\text{sign}(w_{i-1}(j)) \cdot q_i$. This means that the j-th coefficient of the vector w_i is

$$\begin{aligned} w_i(j) &= w_{i-1}(j) - \frac{\|w_{i-1}(j)\|_\infty}{q_i} p_i(j) \\ &= w_{i-1}(j) - \frac{|w_{i-1}(j)|}{q_i} \cdot \text{sign}(w_{i-1}(j)) \cdot q_i = 0. \end{aligned}$$

Hence, the number of non-zero components of w_i is strictly smaller than the number of non-zero components of w_{i-1} and there exists an index $k \leq n$ such that $w_{k+1} = 0$.

It remains to show that for all $1 \leq i \leq k$ the vectors v_i together with the scalars χ_i satisfy the three claimed properties. Obviously, $w = \sum_{i=1}^{k} \chi_i v_i$ and $\chi_i > 0$ for all $1 \leq i \leq k$.

Now, we show that the vectors v_i, $1 \leq i \leq k$, are of small size. To be precise, we will show that for all $1 \leq j \leq n$, we have $|v_i(j)| \leq q_i$. Since the number q_i computed by the revised simultaneous approximation algorithm satisfies $q_i \leq 2^{n^2+n}N^n$ and $v_i(j) \in \mathbb{N}$, it follows that $\text{size}(v_i) \leq 2^{n^2+n}N^n$.
Since the numbers $v_i(j)$ computed by the revised simultaneous approximation algorithm satisfy $|q_i - w'_{i-1}(j) - v_i(j)| \leq 1/N < 1$, they are uniquely determined as the integers which are closest to $q_i \cdot w'_{i-1}(j)$.

- If $|w'_{i-1}(j)| = 1$, it follows from $v_i(j) = w'_{i-1}(j) \cdot q_i$ that $|v_i(j)| = |q_i|$.
- If $|w'_{i-1}(j)| < 1$, then $|q_i w'_{i-1}(j)| < q_i$, where $q_i \in \mathbb{N}$. Hence, $p_i(j)$ as the integer closest to $q_i \cdot w'_{i-1}(j)$ is at most q_i.

For all $1 \leq j \leq k$, the revised simultaneous approximation algorithm with input of the vector $w'_{j-1} \in \mathbb{Q}^n$ and the parameter $N \in \mathbb{N}$ computes an integer $q_j \in \mathbb{N}$ and an integer vector $v_j \in \mathbb{Z}^n$ such that

$$\|q_j \cdot w'_{j-1} - v_j\|_\infty < \frac{1}{N},$$

see Lemma 6.5.3. Since w'_{j-1} is defined by $w_{j-1}/\|w_{j-1}\|_\infty$ and $\chi_j = \|w_{j-1}\|_\infty / q_j$, it follows that

$$\Big\|\frac{1}{\chi_j} w_{j-1} - v_j\Big\|_\infty < \frac{1}{N}.$$

It is easy to see that $w_{j-1} = w - \sum_{i=1}^{j-1} \chi_j v_j = \sum_{i=j}^{k} \chi_i v_i$, which yields

$$\frac{1}{N} > \Big\|\frac{1}{\chi_j} w_{j-1} - v_j\Big\|_\infty = \Big\|\frac{1}{\chi_j}\sum_{i=j}^{k} \chi_i v_i - v_j\Big\|_\infty = \Big\|\sum_{i=j+1}^{k} \frac{\chi_i}{\chi_j} v_i\Big\|_\infty$$

or equivalently

$$\Big\|\sum_{i=j+1}^{k} \chi_i v_i\Big\|_\infty < \frac{\chi_j}{N}. \tag{6.14}$$

By definition of the coefficient χ_j, we have for $1 \leq j \leq k$

$$\chi_{j+1} = \frac{\|w_j\|_\infty}{q_{j+1}} = \frac{\|\sum_{i=j+1}^{k} \chi_i v_i\|_\infty}{q_{j+1}}.$$

Hence, it follows from (6.14) that

$$\frac{\chi_{j+1}}{\chi_j} = \frac{1}{\chi_j} \cdot \frac{\|w_j\|_\infty}{q_{j+1}} = \frac{1}{\chi_j} \cdot \frac{\| \sum_{i=j+1}^k \chi_i v_i\|_\infty}{q_{j+1}} < \frac{1}{\chi_j} \cdot \frac{\chi_j}{N \cdot q_{j+1}} = \frac{1}{N \cdot q_{j+1}}.$$

Since the coefficients of the vector v_j are at most q_{j+1}, we obtain

$$\frac{\chi_{j+1}}{\chi_j} \leq \frac{1}{N\|v_{j+1}\|_\infty}.$$

□

Obviously, if we use Dirichlet's theorem, we can argue in the same way and we obtain that for each vector $w \in \mathbb{Q}^n$, there exists a linear combination of integer vectors with the same properties as in Theorem 6.5.4 except that the size of the integer vectors is at most N^n.

Now, we show that any integer vector whose sum of its coefficients is not too large, is contained in the hyperplane orthogonal to the vector w if and only if it is contained in the hyperplanes orthogonal to the integer vectors v_i, $1 \leq i \leq k$, computed by the decomposition algorithm.

Lemma 6.5.5. *Let $w \in \mathbb{Q}^n \backslash \{0\}$ and $N \in \mathbb{N}$, $N \geq 2$. Let $v_1, \ldots, v_k \in \mathbb{Z}^n$, $k \leq n$, and $\chi_1, \ldots, \chi_k > 0$ be computed by the decomposition algorithm, see Algorithm 17, with input w and N. Thus, they satisfy $w = \sum_{i=1}^k \chi_i v_i$ and for all $2 \leq j \leq k$ we have $\| \sum_{i=j}^k \chi_i v_i\|_\infty < \chi_{j-1}/N$.*
Then, the following holds: For $b \in \mathbb{Z}^n$ with $\|b\|_1 \leq N$ it holds that $b \in H_{0,w}$ if and only if $b \in \bigcap_{i=1}^k H_{0,v_i}$.

Proof. Obviously, every vector $b \in \mathbb{R}^n$ which satisfies $\langle b, v_i \rangle = 0$ for all $1 \leq i \leq k$ is contained in the hyperplane $H_{0,w}$, since $\langle b, w\rangle = \sum_{i=1}^k \chi_i \langle b, v_i \rangle$.
Now, we assume that $b \in \mathbb{Z}^n \cap \bar{B}_n^{(1)}(0, N)$ is contained in the hyperplane $H_{0,w}$. Let $1 \leq j \leq k$ be the smallest index such that $b \notin H_{0,v_j}$, i.e., $0 = \langle w, b\rangle = \langle \sum_{i=j}^n \chi_i v_i, b\rangle$. Obviously, we have

$$\langle \sum_{i=j}^k \chi_j v_j, b\rangle = \chi_j \langle v_j, b\rangle + \sum_{i=j+1}^k \chi_i v_i, b\rangle = \chi_j \left(\langle v_j, b\rangle + \frac{1}{\chi_j} \langle \sum_{i=j+1}^k \chi_i v_i, b\rangle \right).$$

Hence, it follows from Hölder's inequality that

$$\left| \langle v_j, b\rangle - \frac{1}{\chi_j} \langle \sum_{i=j}^k \chi_i v_i, b\rangle \right| = \frac{1}{\chi_j} \left| \langle \sum_{i=j+1}^n \chi_i v_i, b\rangle \right| \leq \frac{1}{\chi_j} \| \sum_{i=j+1}^n \chi_i v_i\|_\infty \cdot \|b\|_1.$$

By assumption, we have $\|\sum_{i=j+1}^{n} \chi_i v_i\|_\infty < \chi_j/N$ and $\|b\|_1 \leq N$, see Theorem 6.5.4. Thus, we obtain that

$$\left| \langle \frac{1}{\chi_j} \sum_{i=j}^{k} \chi_i v_i, b \rangle - \langle v_j, b \rangle \right| < 1. \tag{6.15}$$

Since $b, v_j \in \mathbb{Z}^n$, it follows from $\langle b, v_j \rangle > 0$ that $\langle b, v_j \rangle \geq 1$. Combining this observation with (6.15), we obtain that $(1/\chi_j)\langle \sum_{i=j}^{k} v_i, b \rangle > 0$ or equivalently that $\langle \sum_{i=j}^{k} \chi_i v_i, b \rangle > 0$, which yields a contradiction. □

Replacement procedure

We now present the replacement procedure that replaces a hyperplane $H_{k,d}$ by hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$, with small size. Additionally, the replacement procedure gets an affine subspace H as input. The goal is to secure that any vector from this subspace which is contained in the hyperplane $H_{k,d}$ is also contained in the hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$. Obviously, we cannot guarantee this for all vectors in the affine subspace H but for all integer vectors whose sum of coefficients is not too large, that means for all vectors contained in an ℓ_1-ball with some specific radius.

The idea of the replacement procedure is simple. The algorithm gets an affine subspace H, an additional hyperplane $H_{k,d}$ and a parameter $N \in \mathbb{N}$ as input. The hyperplane is given by a vector $d \in \mathbb{Q}^n$ and a number $k \in \mathbb{Q}$. The algorithm applies the decomposition algorithm to the $(n+1)$-dimensional vector $(d^T, k)^T \in \mathbb{Q}^{n+1}$ and obtains a representation of this vector as a linear combination of integer vectors with small size. These vectors define a set of affine hyperplanes. Using the result from the last section, we can show that all integer vectors with small coefficients which are contained in the original hyperplane are also contained in all new hyperplanes. A concrete description of the algorithm is presented in Algorithm 18.

In the following proposition, we state the main properties of the replacement procedure.

Proposition 6.5.6. *(Proposition 6.1.6 restated.)*
Let $H \subseteq \mathbb{R}^n$ be an affine subspace given by affine hyperplanes H_{k_i,d_i}, $m+1 \leq i \leq n$, and let $H_{k,d}$ be an affine hyperplane such that $d, d_{m+1}, \dots, d_n$ are linearly independent.
Given as input a parameter $N \in \mathbb{N}$, $N \geq 2$, the affine subspace H and the additional affine hyperplane $H_{k,d}$, the replacement procedure, Algorithm 18, computes a set of affinely independent hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J \neq \emptyset$ such that the following holds:

- *Every integer vector $z \in \bar{B}_n^{(1)}(0, N-1) \cap H$ satisfies $\langle d, z \rangle = k$ if and only if it satisfies $\langle \bar{d}_i, z \rangle = \bar{k}_i$ for all $i \in J$.*
- *The affine subspace H and the affine hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$ are affinely independent.*

The size of the vectors $\bar{d}_i \in \mathbb{Z}^n$ and the numbers $\bar{k}_i \in \mathbb{Z}$ is at most $2^{(n+2)^2} N^n$. The number of arithmetic operations of the replacement procedure is at most $(n \cdot \log_2(N))^{\mathcal{O}(1)}$.

Algorithm 18 Replacement procedure

Input:

- A parameter $N \in \mathbb{N}$,
- an affine subspace $H := \bigcap_{i=m+1}^{n} H_{k_i,d_i}$, and
- an additional hyperplane $H_{k,d}$, such that $d, d_{m+1}, \dots, d_n$ are linearly independent.

Used subroutine: Decomposition algorithm.

Output: A collection of hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$.

1. Apply the decomposition algorithm to the vector $w = (d^T, k)^T \in \mathbb{R}^{n+1}$ and the parameter N.
   We obtain vectors $(\bar{d}_i^T, \bar{k}_i)^T \in \mathbb{Z}^{n+1}$ where $1 \leq i \leq j(m) \leq n+1$, together with parameters χ_i, $1 \leq i \leq j(m)$.
2. Let $J \subseteq \{1, \dots, j(m)\}$ be the maximal set of indices such that the vectors d_i, $m+1 \leq i \leq n$ and $\bar{d}_i$, $i \in J$, are linearly independent.
3. Output the affine hyperplanes $H_{\bar{k}_i,\bar{d}_i}$ with $i \in J$.

Proof. First, we show that $J \neq \emptyset$. The decomposition algorithm with input of the vector $(d^T, k)^T$ computes a set of vectors $(\bar{d}_i^T, \bar{k}_i)^T$, $1 \leq i \leq j(m)$. These vectors provide a linear combination of $(d^T, k)^T$,

$$\begin{pmatrix} d \\ k \end{pmatrix} = \sum_{i=1}^{j(m)} \chi_i \begin{pmatrix} \bar{d}_i \\ \bar{k}_i \end{pmatrix}$$

Thus, the vector d is a linear combination of the vectors $\bar{d}_i$. By assumption, the vectors $d_{m+1}, \dots, d_n, d$ are linearly independent. Hence, there exists at least one vector $\bar{d}_i$, $1 \leq i \leq j(m)$, such that the vectors $d_{m+1}, \dots, d_n, \bar{d}_i$ are linearly independent. This guarantees that the subspace H and the affine hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $i \in J$ are affinely independent.

The upper bound on the size follows directly from Theorem 6.5.4, since each vector computed by the decomposition algorithm is an integer vector whose coefficients are at most $2^{(n+1)(n+2)}N^n$, $\|(\bar{d}_i^T, \bar{k}_i)^T\|_\infty \leq 2^{(n+1)(n+2)}N^n$.
Since every integer vector in $z \in \bar{B}_n^{(1)}(0, N-1)$ satisfies $\|z\|_1 \leq N-1$, the vector $z' = (z^T, -1)^T \in \mathbb{Z}^{n+1}$ satisfies $\|z'\|_1 = \|z\|_1 + 1 \leq N$. Hence, it follows from Lemma 6.5.5 that z' is contained in the hyperplane orthogonal to the vector $(d^T, k)^T$ if and only if it is contained in the hyperplanes orthogonal to the vectors $(\bar{d}_i^T, \bar{k}_i)^T$, $1 \leq i \leq j(m)$. This means that the vector z is contained in the hyperplane $H_{k,d}$ if and only if it is contained in the intersection of the hyperplanes $H_{\bar{k}_i,\bar{d}_i}$, $1 \leq i \leq j(m)$, and it shows that

$z \in \bar{B}_n^{(1)}(0, N-1)$ is contained in $H \cap H_{k,d}$ if and only if it is contained in $H \cap \bigcap_{i=1}^{j(m)} H_{\bar{k}_i, \bar{d}_i}$. The set I is maximal with the property that the vectors $d_{m+1}, \ldots, d_n$ and $\bar{d}_i$, $i \in I$, are linearly independent. Hence, a vector $z \in \bar{B}_n^{(1)}(0, N-1)$ is contained in the affine subspace $H \cap H_{k,d}$ if and only if it is contained in the affine subspace $H \cap \bigcap_{i \in I} H_{\bar{k}_i, \bar{d}_i}$. □

This result completes the description of the replacement procedure. Hence, our assumptions made in Section 6.1.3 are satisfied and our lattice membership algorithms presented before are polynomially space bounded.

6.6. Discussion of the results

Overall, we have seen that all assumptions made in the lattice membership algorithm are satisfied. The only thing that remains to be proven is the existence of rounding methods for polytopes and ℓ_p-bodies with $1 < p < \infty$. This will be done in the next chapter.

Except for this aspect we have shown that there exists a polynomially space bounded algorithm that solves the lattice membership problem for all ℓ_p-balls and polytopes exactly. Furthermore, our algorithmic framework can easily be adapted to all classes of full-dimensional bounded convex sets which are closed under bijective affine transformation and intersection with affine hyperplanes if we are able to compute an approximate Löwner-John ellipsoid for each convex set from this class.

The number of arithmetic operations of our lattice membership algorithm is mainly influenced by the factor $n^{(2+o(1))n}$. A substantial improvement of this factor does not seem to be possible. The factor n^{2n} is caused by the fact that in each recursion step, the flatness algorithm computes at most $c \cdot n^2$ affine hyperplanes where we need to search recursively. Here $c \geq 1$ is some fixed constant. The factor $c \cdot n^2$ is comprised of the approximation factor of the computed Löwner-John ellipsoid, which is $c \cdot n$, and the bound n given by the flatness theorem. As we have seen, both results are optimal up to some constant factor.

Dadush, Peikert and Vempala presented an algorithm for the lattice membership problem for well-bounded convex bodies where the number of arithmetic operations is mainly influenced by the factor $n^{(4/3)n}$, where n is the dimension of the convex body, see [DPV11] and [DV12].

The running time of their algorithm is better than ours since they do not approximate the convex body by an ellipsoid as we do. Instead of the *Euclidean version of the flatness theorem*, they use a general version of the flatness theorem which holds for general convex bodies.

Theorem 6.6.1. *Let $K \subseteq \mathbb{R}^n$ be a convex body and $L \subseteq \mathbb{R}^n$. If K does not contain a lattice vector, there exists at most $\mathcal{O}(n^{4/3} \log(n)^c)$ affine hyperplanes $H_{k,d}$ such that K contains a lattice vector from L if and only if there exist one of these hyperplanes such that $K \cap H_{k,d}$ contains a lattice vector from L. Here, $c > 0$ is some fixed constant and* *the vector d is a shortest vector in L^* with respect to the norm defined by the convex body $(K-K)^*$.*

The convex body $K-K$ is the symmetrization of the convex body K, i.e., $K-K = \{x-y|x,y \in K\}$. The dual of a convex body $\mathcal{C} \subseteq \mathbb{R}^n$, denoted by $\mathcal{C}^*$ is the set $\mathcal{C}^* = \{x \in \mathbb{R}^n | \langle x,y\rangle \leq 1$ for all $y \in \mathcal{C}\}$. For a proof of Theorem 6.6.1 see [BLPS99], [Rud00], and [DPV11].

To compute a vector $d \in L^*$ as characterized in Theorem 6.6.1, Dadush, Peikert and Vempala use their single exponential time SVP-algorithm for general norms, which uses single exponential space. Thus they obtain an algorithm for the lattice membership problem which uses single exponential space.

That means, if there exists a polynomially space bounded algorithm that solves the shortest vector problem for some class of norms, one can improve the number of arithmetic operations of the lattice membership algorithm for the class of convex bodies generated by this norm. Of course, the number of arithmetic operations of this algorithm should be at most $n^{(4/3+o(1))n}\log_2(r)^{\mathcal{O}(1)}$, where n is the rank of the lattice and r is an upper bound on its size.

A candidate for such an algorithm is Kannan's algorithm for the shortest vector problem, which we used in our flatness algorithm, see Theorem 4.1.14. As already observed by Kannan, see Remark 2.17 in [Kan87b], this algorithm can easily be generalized to the ℓ_1-norm and the ℓ_∞-norm. Then, the number of arithmetic operations is $\mathcal{O}(3^n n^n \log_2(r))$. If one can generalize this algorithm to the class of all polyhedral norms, one can compute a flatness direction $d \in L^*$ in polynomial space as needed in Theorem 6.6.1. This would lead to a polynomially space bounded algorithm for the lattice membership problem for all polytopes. Particularly, this would yield a polynomially space bounded algorithm for the closest vector problem for the ℓ_∞-norm and the ℓ_1-norm where the number of arithmetic operations is mainly influenced by the factor $n^{(4/3)n}$, where n is the dimension of the polytope.

7. Computation of approximate Löwner-John ellipsoids

The ellipsoid method is an iterative geometric algorithm with polynomial running time that was originally developed by Shor, Yudin and Nemirovskii in the 1970s for the minimization of convex functions, see [Sho77], [YN76a], [YN76b]. In 1979, Khachiyan adapted this method and developed a polynomial time algorithm for linear programming. This was a breakthrough result since linear programming is in $\mathsf{NP} \cap \mathsf{coNP}$ and at that time it was one of the candidates to prove that $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$.

Today, the main impact of the ellipsoid method is not in practice for example for solving linear programming in polynomial time but for its theoretical applications. The ellipsoid method can be used to show the existence of polynomial time algorithms for many geometric and combinatorial optimization problems.

Geometrically, the ellipsoid method can be characterized as a central cut algorithm: In every iteration step we are given an ellipsoid and we have to decide whether one has already found a solution. If this is not the case, we intersect the ellipsoid with an affine hyperplane through the center of the ellipsoid. Already in 1976, Yudin and Nemirovskii remarked that the ellipsoid method does not make full use of the geometric idea behind it. They observed that the number of arithmetic operations of the method remains polynomial if we do not cut the ellipsoid through its center but take more of the original ellipsoid, that means we take a shallow cut. These two types of cuts are illustrated in Figure 7.1.

This modification of the original ellipsoid method, called shallow cut ellipsoid method, allows a number of additional applications. For example, the shallow cut ellipsoid method can be used to compute an approximate Löwner-John ellipsoid of a full-dimensional bounded convex set. To recall, for a parameter $0 < \gamma \leq 1$, a $1/\gamma$-approximate Löwner-John ellipsoid E of a full-dimensional bounded convex set $\mathcal{C}$ is an ellipsoid which is contained in the convex set $\mathcal{C}$, whereas the convex set $\mathcal{C}$ itself is contained in the ellipsoid $(1/\gamma) \star E$, i.e., $E \subseteq \mathcal{C} \subseteq (1/\gamma) \star E$, see Definition 2.2.12 in Chapter 2. John proved that for every full-dimensional bounded convex set there exists an approximate Löwner-John ellipsoid with approximation factor $1/n$, see Theorem 2.2.13 in Chapter 2.

The first algorithm in this area was an algorithm that computes an approximate Löwner-John ellipsoid for polytopes. It was first described by Goffin and extends a method from Lenstra, see [Gof84], [Len83]. More precisely, they present a polyno-

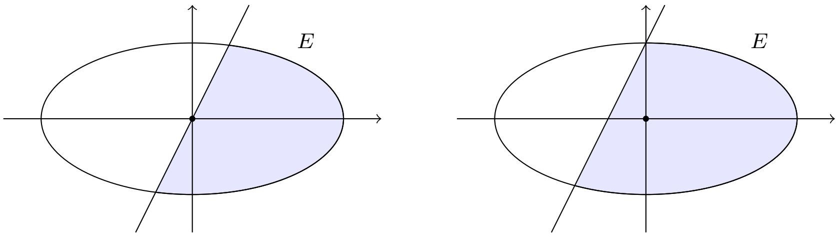

Figure 7.1.: **Two types of cuts of an ellipsoid.** We consider the ellipsoid E centered at the origin with the main axes $(2,0)^T \in \mathbb{R}^2$ and $(0,1)^T \in \mathbb{R}^2$. On the left side, we use the vector $(-2,1)^T$ to cut the ellipsoid through its center. The shadowed region is the intersection $E \cap \{x \in \mathbb{R}^2 | \langle x, (-2,1)^T \rangle \leq 0\}$, a central cut. On the right side, we use the same vector but cut the ellipsoid through the point $(-0.5,0)^T$. Here, the shadowed area is the intersection $E \cap \{x \in \mathbb{R}^2 | \langle x, (-2,1)^T \rangle \leq 1\}$.

mial time algorithm that computes a $2n$-approximate Löwner-John ellipsoid for full-dimensional polytopes, where n is the dimension of the corresponding vector space. This method is also described by Schrijver in [Sch86]. We can use this algorithm to complete the description of the lattice membership algorithm for polytopes presented in Chapter 6 and the description of a deterministic polynomially space bounded algorithm that solves the closest vector problem for all polyhedral norms, in particular for the ℓ_1-norm and the ℓ_∞-norm.

Based on the algorithm of Goffin, Grötschel, Lovász, and Schrijver developed a general algorithmic framework which computes in polynomial time a $\sqrt{n}(n+1)$-approximate Löwner-John ellipsoid. This framework works for all full-dimensional well-bounded convex bodies given by a separation oracle, see [Lov86], [GLS93]. This means, they assume that the algorithm has access to an oracle that decides for a given vector whether it is contained in the convex set or not. If the vector is not contained in the convex set, it provides an affine hyperplane that strictly separates this vector from the convex body.

To use this general framework for concrete convex bodies, one needs to show that these convex bodies are well-bounded, i.e., one needs to compute a circumscribed and an inscribed Euclidean ball for them. Additionally, one need to show that there exists an efficient algorithm that realizes a separation oracle for the given convex bodies.

This work was done by Heinz for convex bodies given by quasiconvex polynomials. He described an algorithm that computes $\mathcal{O}(n^{3/2})$-approximate Löwner-John ellipsoids for convex bodies of the form $Y := \{x \in \mathbb{R}^n | F_i(x) < 0 \text{ for } 0 \leq i \leq s\}$, where the functions $F_i \in \mathbb{Z}[X]$ are quasiconvex polynomials, see [Hei05]. Hildebrand and Köppe improved his algorithm and presented an algorithm that computes for these convex bodies an $\mathcal{O}(n)$-approximate Löwner-John ellipsoid. To improve the approximation factor, they accept that the number of arithmetic operations becomes single exponential in the dimension, see [HK10]. For the improvement, they used an idea of Kochol, who described how an approximate Löwner-John ellipsoid can be computed using the approximation of the Euclidean unit sphere by polytopes, see [Koc94].

In this chapter, we present a general framework that computes a $2/\gamma$-approximate Löwner-John ellipsoid for bounded convex sets which are given by a separation oracle together with a circumscribed Euclidean ball and a lower bound on its volume. The parameter γ needs to satisfy $0 < \gamma < 1/n$. The number of arithmetic operations of this algorithm is polynomial in $1/\gamma$, but single exponential in the dimension n. However, the procedure needs only polynomial space.

Then, we adapt this general framework to the class of ℓ_p-bodies, which we defined in Section 6.4.3 in Chapter 6. The main part here is to show that ℓ_p-bodies are bounded convex sets for which there exists an efficient realization of a separation oracle. Additionally, we need to determine a circumscribed Euclidean ball and a lower bound on the volume. Overall, we achieve an algorithm which computes a $2/\gamma$-approximate Löwner-John ellipsoid for all ℓ_p-bodies with $1 < p < \infty$. This completes the description of the lattice membership algorithm for ℓ_p-bodies presented in Chapter 6 and the description of a deterministic polynomially space bounded algorithm that solves the closest vector problem for all ℓ_p-norms with $1 < p < \infty$.

This chapter is organized as follows. We start with an informal description of the geometric idea behind the ellipsoid method by considering a special case where we are given a bounded convex set by a separation oracle. Additionally, we are given a Euclidean ball which contains the convex set and a lower bound on the volume of the set if it is not empty. The goal is to decide whether the convex set is empty or not.

In Section 7.1, we describe the shallow cut ellipsoid method as a rounding method to compute a $2/\gamma$-approximate Löwner-John ellipsoid for some parameter $0 < \gamma < 1/n$. This method works for all full-dimensional bounded convex sets given by a separation oracle under the assumption that we know a circumscribed Euclidean ball for the convex set and a lower bound on its volume. The number of arithmetic operations of the shallow cut ellipsoid method is single exponential in the dimension, but polynomial in $1/\gamma$.

In the second part of this chapter, we adapt this method to concrete classes of convex sets. In Section 7.2, we consider the class of ℓ_p-bodies. For this class, we show how for a given ℓ_p-body we can compute a circumscribed Euclidean ball, a lower bound on its volume and how we can realize a separation oracle. Unfortunately, we can only guarantee a lower bound on the volume of an ℓ_p-body if the ℓ_p-body contains an integer vector. But this does not matter in our setting.

For the lattice membership algorithm presented in Chapter 6, we also need a rounding method for polytopes. Thus, we also describe the variant of the rounding method which computes for a given full-dimensional polytope $1/\gamma$-approximate Löwner-John ellipsoid for some parameter $0 < \gamma < 1/n$ in polynomial time. Here, our description is based on [Sch86].

The ellipsoid method: an overview

Before we describe the algorithm that computes an approximate Löwner-John ellipsoid, we illustrate the main geometric idea behind the ellipsoid method. To do this, we consider a bounded convex set given by a separation oracle as it is described in Definition 2.1.18 in Chapter 2. The goal is to decide if this convex set is empty or not. If it is non-empty, we want to find a vector in it.

In the following, we will assume that the bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ is given by a separation oracle together with a parameter $v_{in} > 0$. The parameter v_{in} provides a lower bound on the volume of $\mathcal{C}$ if $\mathcal{C}$ is non-empty: If $\mathcal{C}$ is non-empty, then $\mathrm{vol}_n(\mathcal{C}) \geq v_{in}$. Additionally, we assume that we are given an ellipsoid E_0 with center c_0 which contains the convex set, $\mathcal{C} \subseteq E_0$ if $\mathcal{C}$ is non-empty. We can distinguish between two cases:

- Either, the center c_0 is contained in the convex body $\mathcal{C}$, $c_0 \in \mathcal{C}$. Then c_0 is a witness for the fact that $\mathcal{C}$ is non-empty. Whether c_0 is contained in $\mathcal{C}$ can be decided using the separation oracle.

- Or, the center c_0 is not contained in the convex body $\mathcal{C}$. In this case, the idea is to construct a new smaller ellipsoid E_1 which satisfies the following two properties:

  1. the ellipsoid contains the convex body, $\mathcal{C} \subseteq E_1$, and
  2. the volume of the ellipsoid is strictly smaller than the volume of the original ellipsoid E_0 by a factor single exponential in the dimension, that means $\mathrm{vol}_n(E_1) < e^{-1/(c \cdot n)} \mathrm{vol}_n(E_0)$, where $c > 0$ is a constant.

  Such an ellipsoid can be computed in the following way: Since the center c_0 of the original ellipsoid is not contained in the convex set $\mathcal{C}$, there exists an affine hyperplane that separates c_0 from $\mathcal{C}$. Such a hyperplane is given by the separation oracle queried with input of the vector c_0. If this affine hyperplane is given by a vector $a \in \mathbb{R}^n$ we have $\langle a, x \rangle \leq \langle a, c_0 \rangle$ for all $x \in \mathcal{C}$. That means, the convex set is contained in the halfspace

  $$\{x \in \mathbb{R}^n | \langle a, x \rangle \leq \langle a, c_0 \rangle\}.$$

  Together with the assumption that $\mathcal{C} \subseteq E_0$, we get that the convex body is contained in the intersection of the ellipsoid E_0 with this halfspace,

  $$\mathcal{C} \subseteq \{x \in \mathbb{R}^n | \langle a, x \rangle \leq \langle a, c_0 \rangle\} \cap E_0.$$

  Now, we can construct a new ellipsoid E_1 as the smallest ellipsoid that contains this intersection.

Then, the algorithm continues iteratively. Obviously, it outputs the correct answer if it terminates. To guarantee that the algorithm terminates we observe that in each iteration step the volume of the constructed ellipsoid decreases by a single exponential factor $e^{-1/(c \cdot n)}$ for some constant $c > 0$. At the same time, we guarantee that each constructed ellipsoid contains the convex set. This shows that for $\mathcal{C} \neq \emptyset$, the ellipsoid method finds an element $x \in \mathcal{C}$ after at most

$$c \cdot n \cdot \ln\left(\frac{\mathrm{vol}_n(E_0)}{\mathrm{vol}_n(\mathcal{C})}\right)$$

steps of iteration. Together with the fact that we know a lower bound of the volume of $\mathcal{C}$ if $\mathcal{C}$ is not empty, it is easy to see that we can ensure that the ellipsoid method terminates after $\mathcal{O}(n^2) \cdot (\log_2(\mathrm{vol}_n(E_0)) - \log_2(v_{in}))$ steps of iteration.
A more detailed description of the ellipsoid method is given for example in [Sch86], [PS98] or [GLS93].

7.1. The shallow cut ellipsoid method as a method to compute approximate Löwner-John ellipsoids

The basic ellipsoid method always cuts the ellipsoid with an affine hyperplane through the center of the ellipsoid. This divides the ellipsoid into two parts with equal volume. Instead, we consider the intersection of the ellipsoid with a halfspace which contains more than half of the ellipsoid, a shallow cut. We present the shallow cut ellipsoid method as a rounding algorithm in a way such that it computes an approximate Löwner-John ellipsoid.

From now on, we will assume that the convex set $\mathcal{C} \subseteq \mathbb{R}^n$ is full-dimensional and bounded. Additionally, we assume that we have access to a separation oracle $\mathrm{SEP}_{\mathcal{C}}$ for the convex set $\mathcal{C}$ that on input of a vector x decides whether the vector is contained in $\mathcal{C}$ or not. If the vector x is not contained in $\mathcal{C}$, it outputs a vector $a \in \mathbb{R}^n$. This vector defines an affine hyperplane that separates x from $\mathcal{C}$, that means we have $\langle a, x\rangle \geq \langle a, y\rangle$ for all $y \in \mathcal{C}$. Additionally, we assume that the convex set is given together with some parameters $R_{out}, r_{in} \in \mathbb{R}^{>0}$ and a vector $c_0 \in \mathbb{R}^n$ such that

$$\mathcal{C} \subseteq \bar{B}_n^{(2)}(c_{out}, R_{out}) \text{ and } \mathrm{vol}_n(\mathcal{C}) \geq r_{in}^n \cdot \mathrm{vol}_n(B_n^{(2)}(0,1)).$$

The parameter r_{in} provides a lower bound on the volume of the convex body. Later, we will see that it makes sense to parameterize the lower bound in this way. In the rest of this section, whenever we speak of a convex set, we implicitly assume that it is given in this form.

To illustrate the main idea of the rounding method, we consider the situation that we are given a full-dimensional bounded convex set $\mathcal{C}$ together with an ellipsoid E such that $\mathcal{C} \subseteq E$, see Figure 7.2. We have found a $1/\gamma$-approximate Löwner-John ellipsoid of $\mathcal{C}$ for some parameter $0 < \gamma < 1$ if the scaled ellipsoid $\gamma \star E$ is contained in $\mathcal{C}$, $\gamma \star E \subseteq \mathcal{C}$. Thus, the key problem is to check if this is the case. For this, we consider an affine bijective transformation τ, which maps the Euclidean unit ball $\bar{B}_n^{(2)}(0,1)$ to the ellipsoid E. Then, $\gamma \star E \subseteq \mathcal{C}$ if and only if $\bar{B}_n^{(2)}(0,\gamma) \subseteq \tau^{-1}(\mathcal{C})$. Suppose we are given a finite set $\mathcal{N}$ of vectors such that the convex hull of these vectors contains a Euclidean ball with radius γ, $\bar{B}_n^{(2)}(0,\gamma) \subseteq \mathrm{conv}(\mathcal{N})$. Then we can distinguish between two cases:

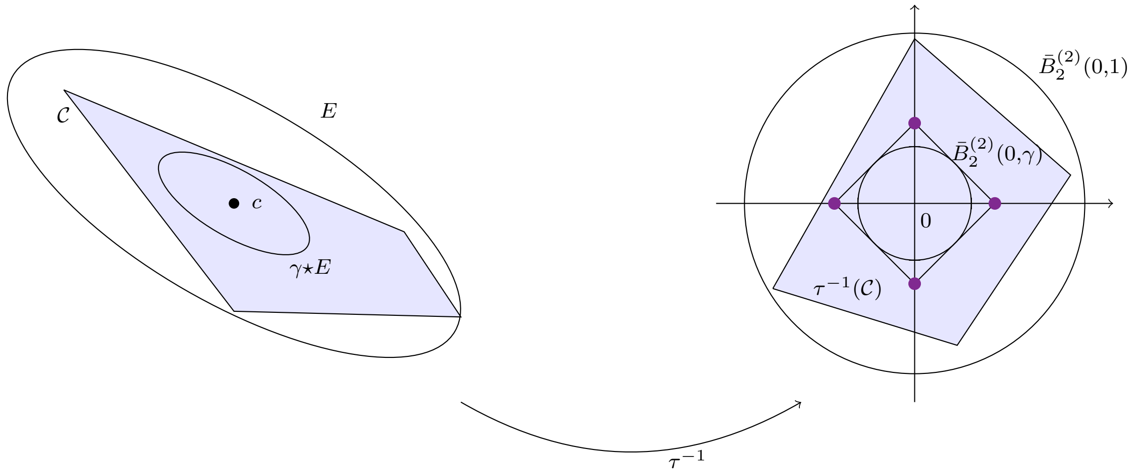

Figure 7.2.: **The main idea of the rounding method.** If the convex set $\tau^{-1}(\mathcal{C})$ contains the set $\mathcal{N} = \{\pm\gamma\sqrt{2}e_i | i = 1, 2\}$, then it contains $\text{conv}(\mathcal{N}) = \bar{B}_2^{(1)}(0, \gamma\sqrt{2})$. Since $\bar{B}_2^{(2)}(0, \gamma) \subseteq \bar{B}_2^{(1)}(0, \gamma\sqrt{2})$, $\tau^{-1}(\mathcal{C})$ contains $\bar{B}_2^{(2)}(0, \gamma)$ and it follows that $\gamma \star E \subseteq \mathcal{C}$.

- If all vectors in $\mathcal{N}$ are contained in $\tau^{-1}(\mathcal{C})$, then it follows from the convexity of $\tau^{-1}(\mathcal{C})$, that also $\bar{B}_n^{(2)}(0, \gamma)$ is contained in $\tau^{-1}(\mathcal{C})$. The decision if $\mathcal{N}$ is contained in $\tau^{-1}(\mathcal{C})$ can be made using the separation oracle for $\mathcal{C}$.

- If there exists a vector in $\mathcal{N}$ which is not contained in $\tau^{-1}(\mathcal{C})$, we use this vector to obtain a halfspace such that the intersection of the halfspace with the ellipsoid E contains the convex set $\mathcal{C}$. Then, we construct a new ellipsoid which contains this intersection and has a smaller volume, and we continue iteratively.

In practice, the proceeding is a little bit different, since we are not able to construct a new ellipsoid with smaller volume if the intersection of the ellipsoid with the halfspace is too large. On this account, we choose a parameter γ with $0 < \gamma < 1/n$ and consider a finite set $\mathcal{N}$ of vectors on the surface of $\bar{B}_n^{(2)}(0, \gamma)$, that means on the sphere $\mathbb{S}^{n-1}(\gamma)$. For these vectors, we check if they are contained in $\tau^{-1}(\mathcal{C})$ or equivalently if their image under the transformation τ is contained in the convex body $\mathcal{C}$, i.e., if $\tau(x) \in \mathcal{C}$ for all $x \in \mathcal{N}$. Then, we distinguish between two cases:

1. If the images of all these elements under the transformation τ are contained in the convex body $\mathcal{C}$,

$$\tau(x) \in \mathcal{C} \text{ for all } x \in \mathcal{N},$$

   we can show that we have found an approximate Löwner-John ellipsoid. The approximation factor depends on the shape of the convex hull of $\mathcal{N}$. If $\text{conv}(\mathcal{N})$

contains a ball with radius $\rho \leq \gamma$ centered at the origin, we can show that we have found a ρ-approximate Löwner-John ellipsoid of $\mathcal{C}$.

2. Otherwise, there exists an element $x \in \mathcal{N}$ whose image under τ is not contained in $\mathcal{C}$, that means, $\tau(x) \notin \mathcal{C}$. In this case, the separation oracle gives us an affine hyperplane that separates the vector $\tau(x)$ from the set $\mathcal{C}$. Thus, we get a vector $a \in \mathbb{R}^n$ such that

$$\langle a, \tau(x)\rangle \geq \langle a, y\rangle \text{ for all } y \in \mathcal{C}.$$

Hence, the convex body $\mathcal{C}$ is fully contained in the intersection of the ellipsoid E with the halfspace $\{x \in \mathbb{R}^n | \langle a, x\rangle \leq \langle a, \tau(\gamma \cdot x)\rangle\}$. Since the parameter γ lies in a certain interval, we can construct a new ellipsoid E_1 which contains this intersection. The new ellipsoid E_1 satisfies two important properties:

- It contains the convex body $\mathcal{C}$ since it contains the intersection of the halfspace with the ellipsoid which itself contains $\mathcal{C}$.
- The volume of E_1 is smaller than the volume of E by a factor single exponential in the dimension n.

We continue iteratively until we find an approximate Löwner-John ellipsoid.

To realize this idea, we need to show two things. Firstly, we need to show how we can construct a set $\mathcal{N} \subseteq \mathbb{S}^{n-1}(\gamma)$ such that $\operatorname{conv}(\mathcal{N})$ contains a ball with large radius $\rho \leq \gamma$. We do this in Section 7.1.1. Secondly, we need to show how such an ellipsoid E_1 satisfying the properties described in 2. can be constructed. This will be done in Section 7.1.2. Then, in Section 7.1.3, we will use these results to give a detailed description and analysis of the algorithm.

7.1.1. Sufficient condition for an approximate Löwner-John ellipsoid

In this section, we consider an ellipsoid $E \subseteq \mathbb{R}^n$ together with an affine bijective transformation $\tau : \mathbb{R}^n \to \mathbb{R}^n$ which maps the Euclidean unit ball to the ellipsoid E. First of all, we show the following: Let $\mathcal{N}$ be a finite set of vectors such that $\operatorname{conv}(\mathcal{N})$ contains a Euclidean ball with radius α centered at the origin. If the images of all elements of $\mathcal{N}$ under the transformation τ are contained in $\mathcal{C}$, then $\mathcal{C}$ contains the shrinked ellipsoid $\alpha \star E$.

Lemma 7.1.1. *Let $E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid given by a symmetric positive definite matrix $D = Q^T Q \in \mathbb{R}^{n\times n}$ and a vector $c \in \mathbb{R}^n$. Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set. We consider the bijective affine transformation $\tau : \mathbb{R}^n \to \mathbb{R}^n$, $x \mapsto Q^T x + c$, i.e., $\tau(\bar{B}_n^{(2)}(0, 1)) = E(D, c)$. Let $\mathcal{N} \subseteq \mathbb{R}^n$ be a finite set with $\bar{B}_n^{(2)}(0, \alpha) \subseteq \operatorname{conv}(\mathcal{N})$ for some $\alpha > 0$. If*

$$\tau(x) \in \mathcal{C} \textit{ for all } x \in \mathcal{N},$$

the ellipsoid $E(D, c)$ scaled by the factor α is contained in $\mathcal{C}$,

$$\alpha \star E(D, c) \subseteq \mathcal{C}.$$

Proof. Since the images of the vectors $x \in \mathcal{N}$ under the transformation τ are contained in $\mathcal{C}$, the vectors x itself are contained in the convex set $\tau^{-1}(\mathcal{C})$, where τ^{-1} is the inverse of the transformation τ, i.e.,

$$x \in \tau^{-1}(\mathcal{C}) \text{ for all } x \in \mathcal{N}.$$

Since $\tau^{-1}(\mathcal{C})$ is convex, it also contains the convex hull of these vectors, $\operatorname{conv}(\mathcal{N}) \subseteq \tau^{-1}(\mathcal{C})$. By assumption, $\operatorname{conv}(\mathcal{N})$ contains a Euclidean ball with radius α. Thus it follows that

$$\alpha \cdot \bar{B}_n^{(2)}(0,1) \subseteq \tau^{-1}(\mathcal{C}).$$

Applying the transformation τ again it follows that

$$\alpha \star E(D,c) \subseteq \mathcal{C}.$$

□

In the rest of this section we will present a concrete construction for the set $\mathcal{N}$ such that the set $\operatorname{conv}(\mathcal{N})$ contains a ball with radius $(1/2-\epsilon)\gamma$ for some $\epsilon > 0$ arbitrary. First, however we present as a motivation the construction of a set $\mathcal{N}$ of size $2n$ which consists of vectors of length $\gamma > 0$ and where the convex hull of $\mathcal{N}$ contains a Euclidean ball with radius $\gamma/\sqrt{n}$.

Corollary 7.1.2. *Let $E(D,c) \subseteq \mathbb{R}^n$ be an ellipsoid given by a symmetric positive definite matrix $D = Q^TQ \in \mathbb{R}^{n\times n}$ and a vector $c \in \mathbb{R}^n$. Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set. We consider the bijective affine transformation $\tau : \mathbb{R}^n \to \mathbb{R}^n,\ x \mapsto Q^Tx+c$. Let $\gamma > 0$ and $e_{n+i} := -e_i$ for $1 \le i \le n$. If*

$$\tau(\gamma \cdot e_i) = c + \gamma Q^T e_i \in \mathcal{C} \textit{ for all } 1 \le i \le 2n,$$

then the ellipsoid $E(D,c)$ scaled by the factor $\gamma/\sqrt{n}$ is contained in $\mathcal{C}$,

$$\frac{\gamma}{\sqrt{n}} \star E(D,c) \subseteq \mathcal{C}.$$

Proof. The proof follows directly, if we apply Lemma 7.1.1 with the set $\mathcal{N} = \{\gamma \cdot e_i | 1 \le i \le 2n\}$. The convex hull of $\mathcal{N}$ is an ℓ_1-ball with radius γ,

$$\operatorname{conv}(\{\gamma \cdot e_i | 1 \le i \le 2n\}) = \bar{B}_n^{(1)}(0,\gamma).$$

It follows from Hölder's inequality that $\bar{B}_n^{(1)}(0,\gamma)$ contains the Euclidean ball $(1/\sqrt{n}) \cdot \bar{B}_n^{(2)}(0,\gamma) = (\gamma/\sqrt{n}) \cdot \bar{B}_n^{(2)}(0,1)$. □

If we would use this result in our algorithm, we would obtain a polynomial time algorithm that computes a $\sqrt{n}/\gamma$-approximate Löwner-John ellipsoid. It is an idea due to Hildebrand and Köppe that the approximation factor can be improved if we consider the convex hull of more than $2n$ vectors from $\bar{B}_n^{(2)}(0,\gamma)$, as it is illustrated in Figure 7.3.

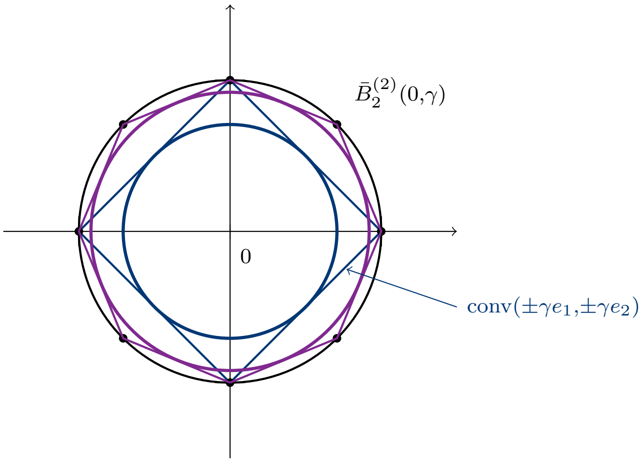
$\bar{B}_2^{(2)}(0,\gamma)$
0
conv($\pm\gamma e_1$,$\pm\gamma e_2$)

Figure 7.3.: **Improvement of the approximation factor.** The square is the convex hull of the main axes of $\mathbb{R}^2$ scaled by the factor γ. The radius of the Euclidean ball contained in this polytope is smaller than the radius of the ball contained in the outer polytope, which is an octagon.

Of course, this mainly influences the number of arithmetic operations of our rounding algorithm. We will obtain an algorithm where the number of arithmetic operations is single exponential in the dimension. But in our context, this does not matter.

The question is, how can we construct the set $\mathcal{N}$ such that it contains a ball with large radius compared with the length of the vectors in $\mathcal{N}$? The idea is to consider a net of $\mathbb{S}^{n-1}(\delta)$, the surface of the ball $\bar{B}_n^{(2)}(0,\delta)$. A δ-net of $\mathbb{S}^{n-1}(\delta)$ is a set $\mathcal{N} \subseteq \mathbb{S}^{n-1}(\delta)$, which provides a covering of the sphere by Euclidean balls with radius δ.

Definition 7.1.3. *(Net of a sphere)*
Let $\delta_1, \delta_2 > 0$. A δ_1-net of $\mathbb{S}^{n-1}(\delta_2) = \{x \in \mathbb{R}^n \mid \|x\|_2 = \delta_2\}$ is a set $\mathcal{N} \subseteq \mathbb{S}^{n-1}(\delta_2)$ such that for every vector $x \in \mathbb{S}^{n-1}(\delta_2)$ there exists a vector $v \in \mathcal{N}$ with Euclidean distance of at most δ_1, i.e., $\|v - x\|_2 \leq \delta_1$.

Kochol observed in [Koc94] that the convex hull of every 1-net of the unit sphere $\mathbb{S}^{n-1} = \mathbb{S}^{n-1}(1)$ contains a Euclidean ball with radius 1/2.

In practice, we are often not able to construct a 1-net of the sphere $\mathbb{S}^{n-1}$ exactly, since we are not able to perform all computations exactly over $\mathbb{R}$. This leads to the approximation of 1-nets. By the ϵ-approximation of a 1-net $\mathcal{N}$, we understand a set $\tilde{\mathcal{N}}$ such that for all $x \in \mathcal{N}$ there exists a vector $\tilde{x} \in \tilde{\mathcal{N}}$ with distance of at most ϵ, i.e., $\|x - \tilde{x}\|_2 \leq \epsilon$. Using the approximation of a 1-net instead of the 1-net itself enables us to use square roots in the construction of a 1-net. Hildebrand and Köppe generalized Kochol's result to ϵ-approximation of 1-nets. The next lemma presents a modified variant of their result, see Lemma 3.2 in [HK10].

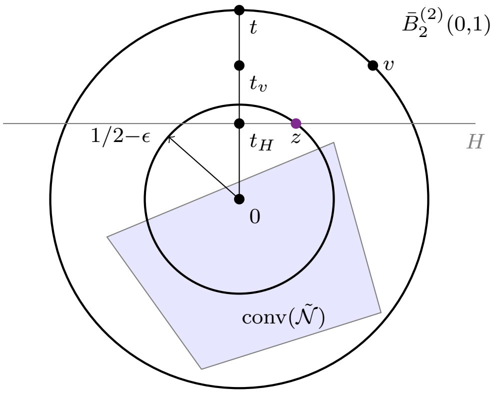

Figure 7.4.: **Illustration of the proof of Lemma 7.1.4.** The affine hyperplane H separates the vector z from $\text{conv}(\tilde{\mathcal{N}})$. The intersection of H with the unit-ball defines a cap with top t.

Lemma 7.1.4. *Let $\mathcal{N}$ be a 1-net of $\mathbb{S}^{n-1}$ and let $0 \leq \epsilon < 1/2$. Suppose that $\tilde{\mathcal{N}}$ is an ϵ-approximation of $\mathcal{N}$, i.e., for all $v \in \mathcal{N}$ there exists $\tilde{v} \in \tilde{\mathcal{N}}$ such that $\|v - \tilde{v}\|_2 \leq \epsilon$. Then, we have*

$$\bar{B}_n^{(2)}\left(0, \frac{1}{2} - \epsilon\right) \subseteq \text{conv}(\tilde{\mathcal{N}}).$$

Proof. The proof of this lemma is illustrated in Figure 7.4. We assume that there exists a vector $z \in \bar{B}_n^{(2)}(0, 1/2-\epsilon)$ which is not contained in the convex hull of $\tilde{\mathcal{N}}$, $z \notin \text{conv}(\tilde{\mathcal{N}})$. Since $\text{conv}(\tilde{\mathcal{N}})$ is convex, there exists an affine hyperplane that strictly separates z from $\text{conv}(\tilde{\mathcal{N}})$, i.e., there exists a vector $p_z \in \mathbb{R}^n$ such that

$$\langle p_z, x\rangle < \langle p_z, z\rangle \text{ for all } x \in \text{conv}(\tilde{\mathcal{N}}).$$

Thus, $\text{conv}(\tilde{\mathcal{N}})$ is completely contained in the halfspace $\{x \in \mathbb{R}^n | \langle p_z, x\rangle < \langle p_z, z\rangle\}$. Now, we consider the cap $\bar{B}_n^{(2)}(0,1) \cap \{x \in \mathbb{R}^n | \langle p_z, x\rangle \geq \langle p_z, z\rangle\}$, which is disjoint from $\text{conv}(\tilde{\mathcal{N}})$. Let $t \in \mathbb{S}^{n-1}$ be the top of this cap, i.e., t is perpendicular to the affine hyperplane

$$H := \{x \in \mathbb{R}^n | \langle p_z, x\rangle = \langle p_z, z\rangle\}.$$

Since $\mathcal{N}$ is a 1-net of $\mathbb{S}^{n-1}$, there exists a vector $v \in \mathcal{N}$ with $\|v - t\|_2 \leq 1$. We will show that the distance between v and the affine hyperplane H is at least ϵ, which yields a contradiction to the fact that $\tilde{\mathcal{N}}$ approximates $\mathcal{N}$.
Let t_v be the orthogonal projection of v on $\text{span}(t)$. Since the distance between t and v is at most 1, we have

$$\|t - t_v\|_2^2 + \|t_v - v\|_2^2 \leq 1. \tag{7.1}$$

Since $v \in \mathbb{S}^{n-1}$, we have

$$\|t_v\|_2^2 + \|t_v - v\|_2^2 = 1. \tag{7.2}$$

Combining (7.1) and (7.2), we obtain

$$\|t - t_v\|_2^2 \leq 1 - \|t_v - v\|_2^2 = \|t_v\|_2^2.$$

Since $\|t_v\|_2 = 1 - \|t - t_v\|_2$, we obtain $\|t - t_v\|_2 = 1 - \|t_v\|_2 < 1 - \|t - t_v\|_2$ or equivalently

$$\|t - t_v\|_2 \leq \frac{1}{2}.$$

Let t_H be the orthogonal projection of t onto H. Since t is perpendicular to H, the vector t_H is also perpendicular to H and it is the point on the affine hyperplane H with minimal distance to the origin. Since $z \in H$ and $\|z\|_2 \leq 1/2 - \epsilon$, this shows that $\|t_H\|_2 \leq 1/2 - \epsilon$. The distance between v and the hyperplane H is $\|t_v - t_H\|_2$, which is at least

$$\|t_v - t_H\|_2 = 1 - \|t - t_v\|_2 - \|t_H\|_2 \geq 1 - \frac{1}{2} - \frac{1}{2} + \epsilon = \epsilon.$$

Combining this with the fact that $\langle p_z, v\rangle > \langle p_z, z\rangle$ and that $\operatorname{conv}(\tilde{\mathcal{N}}) \subseteq \{x \in \mathbb{R}^n | \langle p_z, x\rangle < \langle p_z, z\rangle\}$, this shows that the distance from v to $\operatorname{conv}(\tilde{\mathcal{N}})$ is greater than ϵ. This is a contradiction to the fact that $\tilde{\mathcal{N}}$ approximates $\mathcal{N}$. □

Using this result, we can refine the result of Lemma 7.1.1. Given an ϵ-approximation $\tilde{N}$ of a 1-net of $\mathbb{S}^{n-1}$ the scaled ellipsoid $(1/2 - \epsilon)\gamma \star E(D, c)$ is contained in the convex set $\mathcal{C}$ if the convex hull of $\gamma \star \tilde{N}$ under the transformation τ is contained in the convex set $\mathcal{C}$.

Corollary 7.1.5. *Let $E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid given by a symmetric positive definite matrix $D = Q^T Q \in \mathbb{R}^{n\times n}$ and a vector $c \in \mathbb{R}^n$. Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set. We consider the bijective affine transformation $\tau : \mathbb{R}^n \to \mathbb{R}^n,\ x \mapsto Q^T x + c$. Let $\gamma > 0$ and $0 \leq \epsilon < 1/2$. Let $\tilde{\mathcal{N}} \subseteq \mathbb{Q}^n$ be an ϵ-approximation of a 1-net of $\mathbb{S}^{n-1}$ with $\tilde{\mathcal{N}} \subseteq \bar{B}_n^{(2)}(0, 1)$. If*

$$\tau(\gamma \cdot x) \in \mathcal{C} \textit{ for all } x \in \tilde{\mathcal{N}},$$

the ellipsoid $E(D, c)$ scaled by the factor $(1/2 - \epsilon)\gamma$ is contained in $\mathcal{C}$,

$$\left(\frac{1}{2} - \epsilon\right)\gamma \star E(D, c) \subseteq \mathcal{C}.$$

Proof. The set $\{\gamma \cdot x | x \in \tilde{\mathcal{N}}\}$ is a finite set in the ball $\bar{B}_n^{(2)}(0, \gamma)$. As we have seen in Lemma 7.1.4 the convex hull of $\tilde{N}$ contains the Euclidean ball $\bar{B}_n^{(2)}(0, 1/2 - \epsilon)$. Thus, $\bar{B}_n^{(2)}(0, (1/2-\epsilon)\gamma) \subseteq \operatorname{conv}(\gamma \cdot \tilde{N})$ and the statement follows directly from Lemma 7.1.1. □

If we want to use this result to compute an approximate Löwner-John ellipsoid, we need an explicit construction of a 1-net of the sphere $\mathbb{S}^{n-1}$. Unfortunately, this is not possible with a set whose cardinality is polynomial in the dimension. The size of a 1-net of the unit sphere is at least single exponential in the dimension. In the next lemma, we present an explicit construction of a 1-net of the sphere $\mathbb{S}^{n-1}$. This construction is a slight modification of a construction of Kochol presented in [Koc94]. The size of this net is at most 2^{4n}.

Lemma 7.1.6. *For $n \in \mathbb{N}$, the set*

$$\mathcal{N}_n := \left\{ \frac{x}{\|x\|_2} \;\middle|\; x \in \mathbb{Z}^n \cap \bar{B}_n^{(2)}\left(0, 2\sqrt{n}\right) \setminus \{0\} \right\}$$

is a 1-net on $\mathbb{S}^{n-1}$ with $|\mathcal{N}_n| \leq 2^{4n}$.

Of course, we are not able to compute this 1-net exactly. But, since we have seen in Lemma 7.1.4 that it is also possible to work with the approximation of a 1-net, we will neglect this aspect in the following and we will assume that we are able to compute the 1-net $\mathcal{N}_n$ according to this construction exactly and efficiently[1]. The set of all integer vectors in the Euclidean ball $\bar{B}_n^{(2)}(0, 2\sqrt{n})$ can be computed using a graph-traversal approach like in [MV10a] (see also Proposition 4.2 in [DPV10]). The number of arithmetic operations to do this is $2^{\mathcal{O}(n)}$.

Proof. First we show that $\mathcal{N}_n$ is a 1-net of the sphere $\mathbb{S}^{n-1}$, that means that for every vector $x \in \mathbb{S}^{n-1}$ there exists a vector $v \in \mathcal{N}_n$ whose distance to x is at most 1. For a vector $x \in \mathbb{S}^{n-1}$, we consider the vector $u \in \mathbb{Z}^n$ whose coordinates u_i, $1 \leq i \leq n$, are integers which satisfy

$$2\sqrt{n}|x_i| - 1 < |u_i| \leq 2\sqrt{n}|x_i| \text{ and } \operatorname{sign}(u_i) = \operatorname{sign}(x_i). \tag{7.3}$$

For every $x_i \in \mathbb{R}$ such an integer exists since the interval $(2\sqrt{n}|x_i| - 1, 2\sqrt{n}|x_i|]$ is a half open interval of length 1, which contains exactly one positive integer. The vector $u \in \mathbb{Z}^n$ constructed in this way is unequal to 0. If $u = 0$, we would have $2\sqrt{n}|x_i| < 1$ for all $1 \leq i \leq n$, which yields the contradiction that $\|x\|_2^2 = \sum_{i=1}^n |x_i|^2 < \sum_{i=1}^n 1/(4n) = 1/4$, i.e., $\|x\|_2 < 1/2$. Hence, we have $u \neq 0$.

Since u is defined such that we have $|u_i| \leq 2\sqrt{n}|x_i|$ for all $1 \leq i \leq n$, see (7.3), we have

$$\|u\|_2^2 = \sum_{i=1}^n u_i^2 \leq \sum_{i=1}^n 4nx_i^2 = 4n. \tag{7.4}$$

This shows that $u \in \bar{B}_n^{(2)}(0, 2\sqrt{n})$. Hence, the vector

$$v := u/\|u\|_2 \in \mathcal{N}_n.$$

[1] Of course, efficiently means in time single exponential in the dimension n

Now, we show that the distance between x and v is at most 1. The squared Euclidean distance between these two vectors is

$$\|x-v\|_2^2 = \sum_{i=1}^{n}(x_i - v_i)^2 = \sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i \cdot v_i + \sum_{i=1}^{n} v_i^2 = \|x\|_2^2 - 2\sum_{i=1}^{n} x_i \cdot v_i + \|v\|_2^2.$$

Since $x, v \in \mathbb{S}^{n-1}$, this is

$$\|x-v\|_2^2 = 2\left(1 - \sum_{i=1}^{n} x_i \cdot v_i\right).$$

Thus, to show that $\|x-v\|_2^2 \leq 1$, it suffices to show that $1 - \sum_{i=1}^{n} x_i \cdot v_i \leq 1/2$ or respectively that $\sum_{i=1}^{n} x_i \cdot v_i \geq 1/2$. By definition of v, we have

$$\sum_{i=1}^{n} x_i \cdot v_i = \frac{1}{\|u\|_2}\sum_{i=1}^{n} x_i \cdot u_i = \frac{1}{\|u\|_2}\sum_{i=1}^{n} |x_i| \cdot |u_i|,$$

since $\operatorname{sign}(x_i) = \operatorname{sign}(u_i)$ for all $1 \leq i \leq n$. The coefficients $|u_i|$ are greater than $2\sqrt{n}|x_i| - 1$, see (7.3). Hence, we obtain

$$\begin{aligned}\frac{1}{\|u\|_2}\sum_{i=1}^{n} |x_i| \cdot |u_i| &> \frac{1}{\|u\|_2}\sum_{i=1}^{n} |x_i| \cdot (2\sqrt{n}|x_i| - 1) \\ &= \frac{1}{\|u\|_2}\sum_{i=1}^{n} \left(2\sqrt{n}x_i^2 - |x_i|\right) \\ &= \frac{1}{\|u\|_2}(2\sqrt{n} - \sum_{i=1}^{n} |x_i|),\end{aligned}$$

where the last equality is due to the fact that $x \in \mathbb{S}^{n-1}$. Furthermore, we have $\sum_{i=1}^{n} |x_i| \leq \sqrt{n}$ and it follows that

$$\sum_{i=1}^{n} x_i \cdot v_i > \frac{1}{\|u\|_2}(2\sqrt{n} - \sqrt{n}) = \frac{\sqrt{n}}{\|u\|_2}.$$

We have seen that $\|u\|_2 \leq 2\sqrt{n}$ or respectively that $1/\|u\|_2 \geq 1/(2 \cdot \sqrt{n})$, that means

$$\sum_{i=1}^{n} x_i \cdot v_i > \frac{\sqrt{n}}{2\sqrt{n}} = \frac{1}{2}.$$

This shows that $\mathcal{N}_n$ is a 1-net on the sphere $\mathbb{S}^{n-1}$.

Finally, we show that the net contains at most 2^{4n} elements. Since $|\mathcal{N}_n| \leq |\mathbb{Z}^n \cap \bar{B}_n^{(2)}(0, 2\sqrt{n})|$, we need an upper bound on the number of integer vectors in the ball $\bar{B}_n^{(2)}(0, 2\sqrt{n})$. We could use the standard volume argumentation for lattices here of course, which we presented in Lemma 4.2.11 in Chapter 4. By this, we would obtain an upper bound of $(4\sqrt{n}+1)^n$ but this is too imprecise. We obtain a better result if we consider the special structure of the lattice $\mathbb{Z}^n$. We observe that if we put around each integer vector $x \in \bar{B}_n^{(2)}(0, 2\sqrt{n})$ an open ℓ_∞-ball with radius 1/2, these balls are disjoint,

$$B_n^{(\infty)}\left(x, \frac{1}{2}\right) \cap B_n^{(\infty)}\left(y, \frac{1}{2}\right) = \emptyset \text{ for } x, y \in \mathbb{Z},\ x \neq y.$$

Now we show that $B_n^{(\infty)}(x, 1/2) \subseteq \bar{B}_n^{(2)}(0, (5/2)\cdot\sqrt{n})$ for all $x \in \bar{B}_n^{(2)}(0, 2\sqrt{n})$. It follows from Hölder's inequality that $B_n^{(\infty)}(x, 1/2) \subseteq B_n^{(2)}(x, \sqrt{n}/2)$ for all $x \in \mathbb{R}^n$. For $x \in \bar{B}_n^{(2)}(0, 2\sqrt{n})$, we have

$$B_n^{(\infty)}\left(x, \frac{1}{2}\right) \subseteq \bar{B}_n^{(2)}\left(0, 2\sqrt{n} + \frac{\sqrt{n}}{2}\right) = \bar{B}_n^{(2)}\left(0, \frac{5}{2}\sqrt{n}\right).$$

Hence, the number of integer vectors in the ball $\bar{B}_n^{(2)}(0, 2\sqrt{n})$ is upper bounded by

$$\frac{\operatorname{vol}_n\left(\bar{B}_n^{(2)}(0, \frac{5}{2}\sqrt{n})\right)}{\operatorname{vol}_n\left(B_n^{(\infty)}(0, \frac{1}{2})\right)} = \left(\frac{5}{2}\sqrt{n}\right)^n \operatorname{vol}_n(\bar{B}_n^{(2)}(0, 1)),$$

using that $\operatorname{vol}_n(B_n^{(\infty)}(0, 1/2)) = 1$. We have $\operatorname{vol}_n(B_n^{(2)}(0, 1)) = \pi^{n/2}\left(\Gamma(1+n/2)\right)^{-1}$, where $\Gamma(\cdot)$ denotes the Gamma function, i.e.,

$$|\mathcal{N}_n| = \left(\frac{5}{2}\sqrt{n}\right)^n \pi^{n/2}\left(\Gamma(1+n/2)\right)^{-1}. \tag{7.5}$$

Due to Stirling's formula, see Section A.0.3 in the Appendix, we obtain

$$\Gamma(1+\frac{n}{2}) = \frac{n}{2}\cdot\Gamma(\frac{n}{2}) = \frac{n}{2}\sqrt{2\pi}\left(\frac{n}{2}\right)^{(n-1)/2} e^{-n/2+\nu(n/2)},$$

where ν is a function that satisfies $1 < \nu(n/2) < 6/n$. Obviously, it follows that

$$\Gamma(1+\frac{n}{2}) > \frac{n}{\sqrt{2}}\sqrt{\pi}\sqrt{n}^{n-1}\sqrt{2}^{1-n}e^{-n/2+1} = \sqrt{\pi}\sqrt{2}^{-n} n \cdot \sqrt{n}^{n-1} e^{1-n/2}.$$

Combining this with (7.5), we obtain

$$\begin{aligned}|\mathcal{N}_n| &\leq \left(\frac{5}{2}\sqrt{n}\right)^n \pi^{n/2}\left(\sqrt{\pi}\sqrt{2}^{-n} n\cdot\sqrt{n}^{n-1}e^{1-n/2}\right)^{-1} \\ &= \left(\frac{5}{\sqrt{2}}\right)^n \sqrt{n}n^{-1}\pi^{(n-1)/2}e^{n/2-1} \\ &\leq \left(\frac{5}{\sqrt{2}}\sqrt{\pi\cdot e}\right)^n \leq 16^n = 2^{4n}.\end{aligned}$$

□

7.1.2. Construction of a circumscribed ellipsoid

In this section we are given an ellipsoid $E = E(D, c) \subseteq \mathbb{R}^n$ together with a halfspace H^-. The halfspace $H^- := \{x \in \mathbb{R}^n | \langle a, x\rangle \leq \delta\}$ is given by a vector $a \in \mathbb{R}^n$ and a parameter $\delta \in \mathbb{R}$. We consider the intersection

$$E^- := E \cap H^- = E \cap \{x \in \mathbb{R}^n | \langle a, x\rangle \leq \delta\}$$

of the ellipsoid and the halfspace. Our goal is to construct a new ellipsoid E' which satisfies two properties: For one thing, the ellipsoid E' contains the intersection E^-, $E^- \subseteq E'$. Furthermore, the volume of the ellipsoid E' is smaller than the volume of the ellipsoid E by a single exponential factor.

Before we describe the concrete construction, we consider the halfspace H^- and determine the interval for the parameter δ such that the intersection E^- of the ellipsoid with the halfspace is non-trivial, i.e., neither $E^- = E$ nor $E^- = \emptyset$. The affine hyperplane

$$H := \{x \in \mathbb{R}^n | \langle a, x\rangle = \delta\}$$

has a non-empty intersection with the ellipsoid if and only if

$$\min\{\langle a, x\rangle | x \in E\} \leq \delta \leq \max\{\langle a, x\rangle | x \in E\}.$$

As we have seen in Lemma 6.4.4 in Chapter 6,

$$\max\{\langle a, x\rangle | x \in E\} = \langle a, c\rangle + \sqrt{a^T Da} \text{ and}$$
$$\min\{\langle a, x\rangle | x \in E\} = \langle a, c\rangle - \sqrt{a^T Da},$$

which shows that the affine hyperplane H and the ellipsoid E have a non-empty intersection if and only if

$$\langle a, c\rangle - \sqrt{a^T Da} \leq \delta \leq \langle a, c\rangle + \sqrt{a^T Da}.$$

This condition can be reformulated using an additional parameter ζ. If we consider the following representation of δ as

$$\delta = \langle a, c\rangle + \zeta\sqrt{a^T Da}, \text{ where } \zeta \in \mathbb{R},$$

then the affine hyperplane H and the ellipsoid E have a nontrivial intersection if and only if $-1 \leq \zeta < 1$. With regard to the intersection E^- of the halfspace H^- with the ellipsoid E, this means that E^- is non-empty if the parameter ζ is at least -1. If this intersection is not too large, i.e., if $-1 < \zeta < 1/n$, we call it a shallow cut.

Definition 7.1.7. *(Shallow Cut)*
Let $E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid, $a \in \mathbb{R}^n \backslash \{0\}$, and $\zeta \in \mathbb{R}$. If $-1 \leq \zeta < 1$, the intersection

$$E(D, c) \cap \{x \in \mathbb{R}^n | \langle a, x\rangle \leq \langle a, c\rangle + \zeta\sqrt{a^T Da}\}$$

is called a shallow cut.

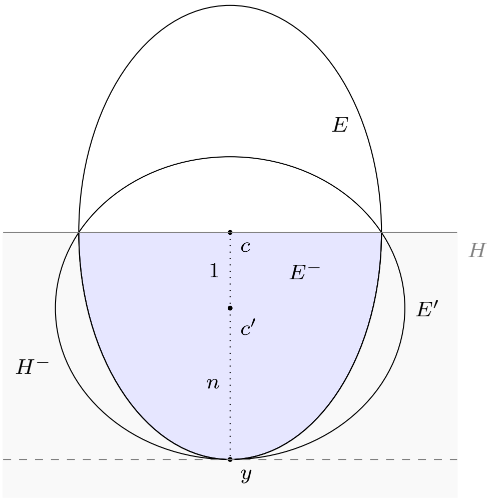

Figure 7.5.: **Construction of an enclosing ellipsoid.** This figure is based on Figure 1.9 in [Sme10].

Given a shallow cut we are able to construct an ellipsoid E' which contains the intersection of the ellipsoid E with the halfspace H^- and whose volume is single exponentially smaller than the volume of E.

To illustrate the main idea behind the construction we assume that we have a central cut, i.e., $\zeta = 0$, see Figure 7.5. Since we consider a central cut, the affine hyperplane H contains the center c of the ellipsoid E and intersects the ellipsoid through its center. The halfspace H^- contains one half of E. Thus there exists an affine hyperplane in H^- parallel to H which supports E, that means the intersection of this affine hyperplane with the ellipsoid consists of a single vector. Let y be this vector. Then, the center c' of the new ellipsoid lies on the segment between c and y and divides this segment into two parts in ratio $1 : n$. Now, the ellipsoid E' is the (unique) ellipsoid with minimal volume centered at c' whose boundary contains y and the intersection $E \cap H$ of the original ellipsoid with the hyperplane H, see [GL81].

Theorem 7.1.8. *Let $E = E(D, c)$ be an ellipsoid in $\mathbb{R}^n$ and $a \in \mathbb{R}^n \backslash \{0\}$. Let $-1 < \zeta < 1/n$. Consider $E^- := E \cap \{x \in \mathbb{R}^n | \langle a, x\rangle \leq \delta\}$, where $\delta := \langle a, c\rangle + \zeta\sqrt{a^T Da}$. Then the ellipsoid $E' = E(D', c')$ with*

$$c' := c - \left(\frac{1 - n\zeta}{n+1} \cdot \frac{1}{\sqrt{a^T Da}}\right) Da$$

and

$$D' := \frac{n^2(1-\zeta^2)}{n^2-1}\left(D - \left(\frac{2}{n+1} \cdot \frac{1-n\zeta}{1-\zeta}\frac{1}{a^T Da}\right) Da(Da)^T\right)$$

satisfies the following properties

1. $E^- \subseteq E'$ *and*
2. $\dfrac{\operatorname{vol}_n(E')}{\operatorname{vol}_n(E)} < e^{-\frac{(1-n\zeta)^2}{2(n+1)}} < 1.$

We will prove this result in two steps. First, we will consider the special case where the ellipsoid is the Euclidean unit ball $\bar{B}_n^{(2)}(0,1)$ and the halfspace is given by the first unit vector $e_1 \in \mathbb{R}^n$. Then we will use the observation that each ellipsoid is the image of the Euclidean unit ball under an affine transformation to show the corresponding result for general ellipsoids. The main part of the proof consists of pure recalculation.
But first of all, we prove a technical statement, where showing how the inverse of the matrix D' defined in Theorem 7.1.8 can be computed.

Lemma 7.1.9. *Let* $D \in \mathbb{R}^{n\times n}$ *be a symmetric positive definite matrix,* $a \in \mathbb{R}^n\backslash\{0\}$ *and* $-1 < \zeta < 1/n$. *Set*

$$D' := \frac{n^2(1-\zeta^2)}{n^2-1}\left(D - \frac{2(1-n\zeta)}{(n+1)(1-\zeta)a^TDa}Da(Da)^T\right).$$

The inverse matrix is

$$D'^{-1} = \frac{n^2-1}{n^2(1-\zeta^2)}\left(D^{-1} + \frac{2(1-n\zeta)}{(n-1)(1+\zeta)a^TDa}aa^T\right).$$

Proof. We show that $D' \cdot D'^{-1} = I_n$. We have

$$\begin{aligned}
&D' \cdot D'^{-1}\\
= &\left(D - \frac{2(1-n\zeta)}{(n+1)(1-\zeta)a^TDa}Da(Da)^T\right) \cdot \left(D^{-1} + \frac{2(1-n\zeta)}{(n-1)(1+\zeta)a^TDa}aa^T\right)\\
= &\ D \cdot D^{-1} + \frac{2(1-n\zeta)}{(n-1)(1+\zeta)}\frac{Daa^T}{a^TDa} - \frac{2(1-n\zeta)}{(n+1)(1-\zeta)}\frac{Da(Da)^TD^{-1}}{a^TDa}\\
&- \frac{2(1-n\zeta)}{(n+1)(1-\zeta)} \cdot \frac{2(1-n\zeta)}{(n-1)(1+\zeta)}\frac{Da(Da)^Taa^T}{(a^TDa)^2}.
\end{aligned}$$

Hence, it is sufficient to show that the sum of the last three summands is zero,

$$\begin{aligned}
&\frac{2(1-n\zeta)}{(n-1)(1+\zeta)}\frac{Daa^T}{a^TDa} - \frac{2(1-n\zeta)}{(n+1)(1-\zeta)}\frac{Da(Da)^TD^{-1}}{a^TDa}\\
&- \frac{2(1-n\zeta)}{(n+1)(1-\zeta)} \cdot \frac{2(1-n\zeta)}{(n-1)(1+\zeta)}\frac{Da(Da)^Taa^T}{(a^TDa)^2}\\
= &\ \frac{1}{a^TDa}\left(\frac{2(1-n\zeta)}{(n-1)(1+\zeta)}Daa^T - \frac{2(1-n\zeta)}{(n+1)(1-\zeta)}Daa^TDD^{-1}\right.\\
&\left.- \frac{4(1-n\zeta)^2}{(n+1)(n-1)(1-\zeta)(1+\zeta)}Da \cdot \frac{a^TDa}{a^TDa} \cdot a^T\right)\\
= &\ \frac{2(1-n\zeta)}{a^TDa \cdot (n+1)(n-1)}\left(\frac{n+1}{1+\zeta} - \frac{n-1}{1-\zeta} - \frac{2(1-n\zeta)}{(1-\zeta)(1+\zeta)}\right)Daa^T.
\end{aligned}$$

It is easy to see that $(n+1)(1-\zeta)-(n-1)(1+\zeta)=2(1-n\zeta)$. From this, it follows that

$$\frac{n+1}{1+\zeta}-\frac{n-1}{1-\zeta}-\frac{2(1-n\zeta)}{(1-\zeta)(1+\zeta)}=0,$$

which shows that the statement is correct. □

Now we prove Theorem 7.1.8 in the special case where the ellipsoid is the Euclidean unit ball $\bar{B}_n^{(2)}(0,1)$ and the affine hyperplane is given by the vector $a=e_1$.

Lemma 7.1.10. *Let $-1<\zeta<1/n$. Consider the intersection $B^- := \bar{B}_n^{(2)}(0,1)\cap\{x\in\mathbb{R}^n | x_1\leq\zeta\}$. Then the ellipsoid $E_B := E(D_B, c_B)$ with*

$$c_B := -\frac{1-n\zeta}{n+1}\cdot e_1$$

and

$$D_B := \frac{n^2(1-\zeta^2)}{n^2-1}\cdot\left(I_n-\frac{2}{n+1}\cdot\frac{1-n\zeta}{1-\zeta}\cdot e_1e_1^T\right),$$

satisfies the following properties

1. *$B^-\subseteq E_B$ and*
2. $\dfrac{\operatorname{vol}_n(E_B)}{\operatorname{vol}_n\left(\bar{B}_n^{(2)}(0,1)\right)}<e^{-\frac{(1-n\zeta)^2}{2(n+1)}}$.

Proof. The proof is achieved by technical calculation.
To prove the first statement, we need to show that every vector from the intersection B^- satisfies $(x-c_B)^T D_B^{-1}(x-c_B)\leq 1$. For this, we observe that the ellipsoid E_B is characterized by the matrix D_B, which is a diagonal matrix of the form

$$\begin{aligned}D_B &= \frac{n^2}{n^2-1}(1-\zeta^2)\left(I_n-\frac{2}{n+1}\cdot\frac{1-n\zeta}{1-\zeta}e_1e_1^T\right)\\ &= \frac{n^2}{n^2-1}(1-\zeta^2)\operatorname{diag}\left(1-\frac{2}{n+1}\cdot\frac{1-n\zeta}{1-\zeta},1,\ldots,1\right).\end{aligned}$$

Obviously, D_B is symmetric positive definite. Since

$$\begin{aligned}\frac{n^2}{n^2-1}(1-\zeta^2)\cdot\left(1-\frac{2(1-n\zeta)}{(n+1)(1-\zeta)}\right) &= \frac{n^2}{n^2-1}(1-\zeta^2)\cdot\frac{(n-1)\cdot(1+\zeta)}{(n+1)(1-\zeta)}\\ &= \frac{n^2}{(n+1)^2}(1+\zeta)^2,\end{aligned}$$

the inverse matrix of the matrix D_B is

$$D_B^{-1}=\operatorname{diag}\left(\frac{(n+1)^2}{n^2(1+\zeta)^2},\frac{n^2-1}{n^2(1-\zeta^2)},\ldots,\frac{n^2-1}{n^2(1-\zeta^2)}\right).$$

We can show that for all $x \in \mathbb{R}^n$, we have

$$\begin{aligned} & (x - c_B)^T D_B^{-1} (x - c_B) \\ = \; & \frac{n^2 - 1}{n^2(1-\zeta^2)} \left(\|x\|_2^2 - 1\right) + 2\frac{(n+1)(1-\zeta n)}{n^2(1+\zeta)^2(1-\zeta)}(x_1 - \zeta)(x_1 + 1) + 1. \end{aligned} \tag{7.6}$$

The proof of this statement is very technical and can be found at the end of this section, see Claim 7.1.12. To show that the term (7.6) is at most 1, we consider the first two summands and show that they are not positive if $x \in \bar{B}^{(2)}(0,1) \cap \{x \in \mathbb{R}^n | x_1 \leq \zeta\}$.

- Since $|\zeta| < 1$, we get that $(n^2-1)/n^2(1-\zeta^2) > 0$. As x is contained in the ℓ_2-unit ball, we have $\|x\|_2^2 - 1 \leq 0$. Combining these two observations, we obtain

$$\frac{n^2-1}{n^2(1-\zeta^2)}(\|x\|_2^2 - 1) < 0.$$

- Since $\zeta < 1/n$, we have $1 - \zeta n \geq 0$ and $(n+1)(1-\zeta n)/(n^2(1+\zeta)^2(1-\zeta)) > 0$. From $-1 \leq x_1 \leq \zeta$, it follows that $x_1 - \zeta \leq 0$ and $x_1 + 1 \geq 0$. Hence, we get that

$$2\frac{(n+1)(1-\zeta n)}{n^2(1+\zeta)^2(1-\zeta)}(x_1 - \zeta)(x_1 + 1) \leq 0.$$

This shows that the first two summands in (7.6) are at most 0 and we obtain $(x - c')^T D'^{-1}(x - c') \leq 1$. It remains to show that the volume of the ellipsoid E_B is smaller than the volume of the unit ball by a single exponential factor. Since

$$\mathrm{vol}_n(E_B) = \sqrt{\det(D_B)} \cdot \mathrm{vol}_n(B_n^{(2)}(0,1)),$$

see Lemma 2.2.8 in Chapter 2, the ratio of the volume of the ellipsoid E_B and the volume of the ball $\bar{B}_n^{(2)}(0,1)$ is

$$\begin{aligned} \left(\frac{n^2(1+\zeta)^2}{(n+1)^2} \cdot \prod_{i=1}^{n-1} \frac{n^2(1-\zeta^2)}{n^2-1}\right)^{1/2} &= \frac{n(1+\zeta)}{n+1} \cdot \left(\frac{n^2(1-\zeta^2)}{n^2-1}\right)^{(n-1)/2} \\ &= \left(1 - \frac{1 - n\cdot\zeta}{n+1}\right) \cdot \left(1 + \frac{1-n^2\zeta^2}{n^2-1}\right)^{(n-1)/2}. \end{aligned}$$

Using that $1 + x \leq e^x$ for all $x \in \mathbb{R}$, we get

$$\begin{aligned} \frac{\mathrm{vol}_n(E_B)}{\mathrm{vol}_n\left(\bar{B}_n^{(2)}(0,1)\right)} &\leq e^{-\frac{1-n\cdot\zeta}{n+1}} \cdot e^{\frac{1-n^2\zeta^2}{n^2-1}\cdot\frac{n-1}{2}} \\ &= e^{-\frac{(1-n\zeta)^2}{2(n+1)}}. \end{aligned}$$

□

To prove Theorem 7.1.8 we need to transfer this result to arbitrary ellipsoids. Thereby it is not enough to consider any bijective affine transformation which maps the Euclidean unit ball to the ellipsoid $E(D, c)$. We need a transformation with the additional property that it maps the constructed ellipsoid E_B to the ellipsoid E' and B^- to E^-. The existence of such a transformation is proven in the following lemma.

Lemma 7.1.11. *Let $E = E(D, c) \subseteq \mathbb{R}^n$ be an ellipsoid and $a \in \mathbb{R}^n \backslash \{0\}$. Let $-1 < \zeta < 1/n$. We consider the ellipsoid E' and the set E^- defined as in Theorem 7.1.8 and the ellipsoid E_B and the set E^- defined as in Lemma 7.1.10. Then there exists a bijective affine transformation $\bar{\tau} : \mathbb{R}^n \to \mathbb{R}^n$ which maps the Euclidean unit ball to the ellipsoid $E(D, c)$ and satisfies the following properties:*

- *The transformation maps the intersection from the unit ball with the halfspace $\{x \in \mathbb{R}^n | \langle e_1, x\rangle \leq \zeta\}$ to the intersection of the ellipsoid with the halfspace $\{x \in \mathbb{R}^n | \langle a, x\rangle \leq \delta\}$, with $\delta = \langle a, x\rangle + \zeta\sqrt{a^T Da}$, i.e., $\bar{\tau}(B^-) = E^-$, and*
- *the transformation maps the circumscribed ellipsoid E_B to the ellipsoid E', that means $\bar{\tau}(E_B) = E'$.*

Proof. The transformation is characterized such that it maps the center of the ellipsoid E_B to the center of the ellipsoid E'. The center of the ellipsoid E_B is the vector

$$c_B = -\frac{1 - n\zeta}{n+1} e_1.$$

Since $-1 < \zeta < 1/n$, we have $|(1 - n\zeta)/(n + 1)| < 1$ and $c_B \in \bar{B}_n^{(2)}(0, 1)$. The center of the ellipsoid E' is defined as

$$c' = c - \frac{1 - n\zeta}{n+1} \frac{Da}{\sqrt{a^T Da}}.$$

In Lemma 2.2.9 in Chapter 2, we have seen that there exists a bijective affine transformation from the Euclidean unit ball to the ellipsoid E which maps c_B to c' if $c_B^T c_B = (c' - c)^T D^{-1}(c' - c)$. Since

$$\begin{aligned}
&(c - \frac{1 - n\zeta}{n+1} \frac{Da}{\sqrt{a^T Da}} - c)^T D^{-1}(c - \frac{1 - n\zeta}{n+1} \frac{Da}{\sqrt{a^T Da}} - c) \\
&= (\frac{1 - n\zeta}{n+1})^2 \frac{1}{a^T Da} (Da)^T D^{-1} Da \\
&= (\frac{1 - n\zeta}{n+1})^2 \frac{1}{a^T Da} a^T Da = (\frac{1 - n\zeta}{n+1})^2 < 1,
\end{aligned}$$

the vector c' is contained in the ellipsoid $E(D, c)$ and it holds that

$$(c' - c)^T D^{-1}(c' - c) = c_B^T \cdot c_B.$$

Hence, we are able to define a bijective affine transformation $\bar{\tau} : x \mapsto Q^T x + c$ where $D = Q^T Q$ such that $\bar{\tau}(\bar{B}_n^{(2)}(0, 1)) = E(D, c)$ and

$$\bar{\tau}(c_B) = c'. \tag{7.7}$$

It remains to show that this transformation satisfies the stated properties. To do this, we observe that the property (7.7) can be rewritten as follows

$$Q^T(-\frac{1-n\zeta}{n+1}e_1)+c=c'=c-\frac{1-n\zeta}{n+1}\frac{Da}{\sqrt{a^TDa}}$$

or equivalently that

$$a=\sqrt{a^TDa}\cdot D^{-1}Q^Te_1=\sqrt{a^TDa}\cdot Q^{-1}e_1. \tag{7.8}$$

With this observation, the rest of the proof are merely technical computations.

- To prove that $\bar{\tau}(B^-)=E^-$, we consider an arbitrary vector $x\in B^-=\bar{B}_n^{(2)}(0,1)\cap\{x\in\mathbb{R}^n|x_1\leq\zeta\}$. Since $\bar{\tau}$ maps the Euclidean unit ball to the ellipsoid E, it follows from $x\in\bar{B}_n^{(2)}(0,1)$ that $\bar{\tau}(x)\in E(D,c)$. To show that $\bar{\tau}(x)\in\{x\in\mathbb{R}^n|\langle a,x\rangle\leq\delta\}$, we observe that it follows from (7.8) that

  $$\begin{aligned}\langle\bar{\tau}(x),a\rangle&=\langle Q^Tx+c,a\rangle=\langle c,a\rangle+\sqrt{a^TDa}\cdot\langle Q^Tx,Q^{-1}e_1\rangle\\&=\langle c,a\rangle+\sqrt{a^TDa}\cdot\langle x,e_1\rangle.\end{aligned}$$

  Since $x\in\{y\in\mathbb{R}^n|\langle y,e_1\rangle\leq\zeta\}$, this is at most

  $$\langle\bar{\tau}(x),a\rangle\leq\langle c,a\rangle+\sqrt{a^TDa}\langle x,e_1\rangle\leq\langle c,a\rangle+\zeta\sqrt{a^TDa}$$

  which means that

  $$\bar{\tau}(x)\in E^-=E(D,c)\cap\{x\in\mathbb{R}^n|\langle a,x\rangle\leq\delta\}$$

  with $\delta=\langle a,x\rangle+\zeta\sqrt{a^TDa}$.

- To show that $\bar{\tau}(E_B)=E'$, we consider a vector $x\in E_B$. Our goal is to show that $\bar{\tau}(x)\in E'$, i.e., $(Q^Tx+c-c')^TD'^{-1}(Q^Tx+c-c')\leq1$.
  Since the transformation $\bar{\tau}$ is defined such that $\bar{\tau}(c_B)=Q^Tc_B+c=c'$, we obtain that

  $$c'-c=Q^Tc_B. \tag{7.9}$$

  If we can show that $QD'^{-1}Q^T=D_B^{-1}$, then we obtain that

  $$\begin{aligned}(Q^Tx+c-c')^TD'^{-1}(Q^Tx+c-c')&=(Q^Tx-Q^Tc_B)^TD'^{-1}(Q^Tx-Q^Tc_B)\\&=(x-c_B)^TQD'^{-1}Q^T(x-c_B)\\&=(x-c_B)^TD_B^{-1}(x-c_B)\leq1,\end{aligned}$$

  where the last inequality follows from the assumption that $x\in E(D_B,c_B)$. Hence, it remains to show that $D_B^{-1}=QD'^{-1}Q^T$. According to Lemma 7.1.9, we have

  $$QD'^{-1}Q^T=\frac{n^2-1}{n^2(1-\zeta^2)}\left(QD^{-1}Q^T+\frac{2(1-n\zeta)}{(n-1)(1+\zeta)}\frac{1}{a^TDa}Qaa^TQ^T\right)$$

and

$$D_B^{-1} = \frac{n^2-1}{n^2(1-\zeta^2)}(I_n + \frac{2(1-n\zeta)}{(n-1)(1+\zeta)}e_1e_1^T).$$

So, we need to show that $QD'^{-1}Q^T = I_n$ and that $(1/a^TDa)\cdot Qaa^TQ^T = e_1e_1^T$. The first statement is obvious since Q is a decomposition of D. The second statement follows from $Qa/\sqrt{a^TDa} = e_1$, see Equation (7.8).

□

Using this result, the proof of Theorem 7.1.8 follows directly from Lemma 7.1.11, together with the fact that the relation between the volumes of two ellipsoids is invariant under affine bijective transformation see Equation (2.2) on page 27 in Chapter 2.
The only thing that remains to be proven is the following statement.

Claim 7.1.12. *For $-1 < \zeta < 1/n$, let*

$$D_B^{-1} = \operatorname{diag}\left(\frac{(n+1)^2}{n^2(1+\zeta)^2}, \frac{n^2-1}{n^2(1-\zeta^2)}, \dots, \frac{n^2-1}{n^2(1-\zeta^2)}\right)$$

and

$$c_B = -\frac{1-n\zeta}{n+1}e_1.$$

For all $x \in \mathbb{R}^n$ we have that

$$\begin{aligned} &(x-c_B)^TD_B^{-1}(x-c_B) \\ = \; &\frac{n^2-1}{n^2(1-\zeta^2)}\left(\|x\|_2^2 - 1\right) + 2\frac{(n+1)(1-\zeta n)}{n^2(1+\zeta)^2(1-\zeta)}(x_1-\zeta)(x_1+1) + 1. \end{aligned}$$

Proof. For $x \in \mathbb{R}^n$ we have

$$x - c_B = \left(x_1 + \frac{1-n\zeta}{n+1}\right)e_1 + \sum_{i=2}^n x_ie_i.$$

Since D_B^{-1} is a diagonal matrix, it follows that

$$(x-c_B)^TD_B^{-1}(x-c_B) = \left(x_1 + \frac{1-n\zeta}{n+1}\right)^2\frac{(n+1)^2}{n^2(1+\zeta)^2} + \sum_{i=2}^n x_i^2\frac{n^2-1}{n^2(1-\zeta^2)}$$

with

$$\begin{aligned} &\left(x_1 + \frac{1-n\zeta}{n+1}\right)^2\frac{(n+1)^2}{n^2(1+\zeta)^2} \\ = \; &x_1^2\frac{(n+1)^2}{n^2(1+\zeta)^2} + 2x_1\frac{1-n\zeta}{n+1}\cdot\frac{(n+1)^2}{n^2(1+\zeta)^2} + \frac{(1-n\zeta)^2}{(n+1)^2}\cdot\frac{(n+1)^2}{n^2(1+\zeta)^2} \\ = \; &x_1^2\frac{n^2-1}{n^2(1-\zeta^2)} + x_1^2\left(\frac{(n+1)^2}{n^2(1+\zeta)^2} - \frac{n^2-1}{n^2(1-\zeta^2)}\right) + 2x_1\frac{(1-n\zeta)(n+1)}{n^2(1+\zeta)^2} + \frac{(1-n\zeta)^2}{n^2(1+\zeta)^2} \\ = \; &x_1^2\frac{n^2-1}{n^2(1-\zeta^2)} + x_1^2\left(\frac{2(n+1)(1-n\zeta)}{n^2(1+\zeta)^2(1-\zeta)}\right) + 2x_1\frac{(1-n\zeta)(n+1)}{n^2(1+\zeta)^2} + \frac{(1-n\zeta)^2}{n^2(1+\zeta)^2}. \end{aligned}$$

This shows that

$$\begin{aligned}
&(x-c_B)^T D_B^{-1}(x-c_B)\\
= \ & \frac{n^2-1}{n^2(1-\zeta^2)}\|x\|_2^2 + x_1^2\left(\frac{2(n+1)(1-n\zeta)}{n^2(1+\zeta)^2(1-\zeta)}\right) + 2x_1\frac{(1-n\zeta)(n+1)}{n^2(1+\zeta)^2} + \frac{(1-n\zeta)^2}{n^2(1+\zeta)^2}.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
\frac{(1-n\zeta)^2}{n^2(1+\zeta)^2} &= -\frac{n^2-1}{n^2(1-\zeta^2)} + \frac{(1-n\zeta)^2}{n^2(1+\zeta)^2} + \frac{n^2-1}{n^2(1-\zeta^2)}\\
&= -\frac{n^2-1}{n^2(1-\zeta^2)} + \frac{-2n\zeta+n^2\zeta^2-2\zeta+2n\zeta^2-n^2\zeta^3+n^2+n^2\zeta}{n^2(1+\zeta)(1-\zeta^2)},
\end{aligned}$$

that means

$$\begin{aligned}
&(x-c_B)^T D_B^{-1}(x-c_B)\\
= \ & \frac{n^2-1}{n^2(1-\zeta^2)}\left(\|x\|_2^2-1\right) + x_1^2\left(\frac{2(n+1)(1-n\zeta)}{n^2(1+\zeta)^2(1-\zeta)}\right) + 2x_1\frac{(1-n\zeta)(n+1)}{n^2(1+\zeta)^2}\\
& + \frac{-2n\zeta+n^2\zeta^2-2\zeta+2n\zeta^2-n^2\zeta^3+n^2+n^2\zeta}{n^2(1+\zeta)(1-\zeta^2)}.
\end{aligned}$$

It holds that

$$\begin{aligned}
& \frac{-2n\zeta+n^2\zeta^2-2\zeta+2n\zeta^2-n^2\zeta^3+n^2+n^2\zeta}{n^2(1+\zeta)(1-\zeta^2)}\\
= \ & \frac{-2n\zeta+n^2\zeta^2-2\zeta+2n\zeta^2-n^2\zeta^3+n^2+n^2\zeta-n^2(1+\zeta)(1-\zeta^2)}{n^2(1+\zeta)(1-\zeta^2)}+1\\
= \ & 2\zeta\frac{-n+n^2\zeta-1+n\zeta}{n^2(1+\zeta)(1-\zeta^2)}+1\\
= \ & -2\zeta\frac{(n+1)(1-n\zeta)}{n^2(1+\zeta)(1-\zeta^2)}+1.
\end{aligned}$$

Combing all this, we obtain

$$\begin{aligned}
&(x-c_B)^T D_B^{-1}(x-c_B)\\
= \ & \frac{n^2-1}{n^2(1-\zeta^2)}\left(\|x\|_2^2-1\right) + x_1^2\left(\frac{2(n+1)(1-n\zeta)}{n^2(1+\zeta)^2(1-\zeta)}\right) + 2x_1\frac{(1-n\zeta)(n+1)}{n^2(1+\zeta)^2}\\
& -2\zeta\frac{(n+1)(1-n\zeta)}{n^2(1+\zeta)(1-\zeta^2)}+1\\
= \ & \frac{n^2-1}{n^2(1-\zeta^2)}\left(\|x\|_2^2-1\right) + \frac{2(n+1)(1-n\zeta)}{n^2(1+\zeta)(1-\zeta^2)}\left(x_1^2+2x_1(1-\zeta)+1\right)+1\\
= \ & \frac{n^2-1}{n^2(1-\zeta^2)}\left(\|x\|_2^2-1\right) + \frac{2(n+1)(1-n\zeta)}{n^2(1+\zeta)(1-\zeta^2)}(x_1+1)(x_1-\zeta)+1.
\end{aligned}$$

□

7.1.3. Description and analysis of the rounding procedure for bounded convex sets

Using the previous results, we are able to present a method that computes approximate Löwner-John ellipsoids for full-dimensional bounded convex sets given by a separation oracle. We call this procedure a rounding algorithm for convex sets.

The input of the algorithm is a full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ given by a separation oracle $\mathrm{SEP}_{\mathcal{C}}$. Additionally, we get a vector $c_{out} \in \mathbb{R}^n$ and a parameter $R_{out} > 0$ such that $\mathcal{C} \subseteq \bar{B}_n^{(2)}(c_{out}, R_{out})$. This ball is used as the initial ellipsoid. Furthermore, the input is a parameter $r_{in} > 0$, which provides a lower bound on the volume of the convex set $\mathcal{C}$, $\mathrm{vol}_n(\mathcal{C}) \geq r_{in}^n \,\mathrm{vol}_n(B_n^{(2)}(0,1))$.

After the initialization, the algorithm works iteratively. Given an ellipsoid $E(D_k, c_k)$, it computes a decomposition $D_k = Q_k^T Q_k$ of the matrix D_k. This decomposition defines a bijective affine transformation τ_k from the Euclidean unit ball to the ellipsoid $E(D_k, c_k)$, $\tau_k : x \mapsto Q_k^T x + c_k$.

Then, we consider a 1-net $\mathcal{N}$ of the sphere $\mathbb{S}^{n-1}$. We construct this net $\mathcal{N}$ according to the construction presented in Lemma 7.1.6. For each element $\gamma \cdot x$ with $x \in \mathcal{N}$, we consider its image under the transformation τ_k. That means, the algorithm checks if the vectors $\tau_k(\gamma \cdot x)$, $x \in \mathcal{N}$, are contained in the convex body $\mathcal{C}$. If all these vectors are contained in $\mathcal{C}$ then the ellipsoid $(\gamma/2) \star E(D_k, c_k)$ is contained in $\mathcal{C}$, as we have seen in Lemma 7.1.5 and the algorithm outputs the ellipsoid $(\gamma/2) \star E(D_k, c_k)$.
Otherwise, there exists an element $x \in \mathcal{N}$ such that $\tau(\gamma \cdot x)$ is not contained in the convex set $\mathcal{C}$. In this case, the separation oracle queried with input of the vector $\tau_k(\gamma \cdot x)$ gives a separating hyperplane. That means, it outputs a vector $a \in \mathbb{Q}^n$ such that

$$\langle a, \tau_k(x) \rangle \geq \langle a, x \rangle \text{ for all } x \in \mathcal{C}.$$

Now, the algorithm considers the intersection of the ellipsoid E_k with the halfspace $\{x \in \mathbb{R}^n | \langle a, x \rangle \leq \langle a, \tau_k(x) \rangle\}$ and constructs an ellipsoid E_{k+1} which contains this intersection. Since we have scaled the unit ball by a factor $\gamma < 1/n$, we have a shallow cut and are able to construct the ellipsoid E_{k+1} according to the construction in Section 7.1.2. The algorithm terminates after at most N iterations, the parameter N will be defined later. Of course, we use the same net $\mathcal{N}$ in each iteration-step but we need to compute it explicitly in each step if we want to guarantee that the algorithm runs in polynomial space. A complete description of the algorithm is given in Algorithm 19.

Now, we will analyze the algorithm. During its execution, the algorithm computes a number of ellipsoids $E_k = E(D_k, c_k)$. The number of computed ellipsoids depends on the moment of termination of the algorithm, but it is upper bounded by N. In the following, if we speak of a constructed ellipsoid E_k, we assume that the algorithm is running through at least k iterations and Step 2(b)ii is executed at least k-times.

Algorithm 19 Rounding algorithm for convex sets

```
Input:
  • A full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ given by a separation oracle $\mathrm{SEP}_{\mathcal{C}}$,
  • parameters $r_{in}, R_{out} > 0$, a vector $c_{out} \in \mathbb{R}^n$, and
  • a parameter $\gamma$ with $0 < \gamma < 1/n$.
Output: an ellipsoid $E \subseteq \mathbb{R}^n$ given by a symmetric positive definite matrix $D$ and a center $c$.
  1. (Initialization)
     Set
       • $N \leftarrow \left\lceil 2\frac{(n+1)n}{(1-n\gamma)^2}\left(\log_2(R_0) - \log_2(r_{in})\right)\right\rceil$,
       • $D_0 \leftarrow R_{out}^2 \cdot I_n$, and $c_0 \leftarrow c_{out}$.
  2. For $0 \leq k \leq N$,
     a) compute a decomposition of the matrix $D_k = Q_k^T Q_k$.
     b) Check if there exists an element $x \in \{x/\|x\|_2 | x \in \mathbb{Z}^n \cap \bar{B}_n^{(2)}(0, 2\sqrt{n}) \backslash \{0\}\}$ such that
          $$c_k + \gamma Q_k^T x \notin \mathcal{C}.$$
        i. If no such element exists, output $E((\gamma^2/4) \cdot D_k, c_k)$.
        ii. Otherwise, query $\mathrm{SEP}_{\mathcal{C}}$ with input of the vector $c_k + \gamma Q_k^T x$. The result is a vector $a \in \mathbb{R}^n \backslash \{0\}$. Set
          $$c_{k+1} \leftarrow c_k - \frac{1-n\gamma}{n+1} \cdot \frac{D_k a}{\sqrt{a^T D_k a}} \text{ and}$$
          $$D_{k+1} \leftarrow \frac{n^2(1-\gamma^2)}{n^2-1}\left(D_k - \frac{2(1-n\gamma)}{(n+1)(1-\gamma)} \cdot \frac{D_k a (D_k a)^T}{a^T D_k a}\right).$$
```

In the next lemma, we state the main properties of the rounding method for convex sets. We will show that each ellipsoid constructed by the algorithm satisfies the property that it contains the convex set. Additionally, we will show that the output of the algorithm is an approximate Löwner-John ellipsoid.

Lemma 7.1.13. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional bounded convex set given by a separation oracle* $\mathrm{SEP}_{\mathcal{C}}$*. Let $R_{out} > 0$ and $c_{out} \in \mathbb{R}^n$ such that $\mathcal{C} \subseteq \bar{B}_n^{(2)}(c_{out}, R_{out})$ and $0 < \gamma < 1/n$. Then the rounding algorithm for convex sets, Algorithm 19, satisfies the following properties:*

- *Each ellipsoid E_k constructed by the algorithm contains the convex set, $\mathcal{C} \subseteq E_k$ for all $k \geq 0$.*
- *The output of the algorithm is a $2/\gamma$-approximate Löwner-John ellipsoid, that means an ellipsoid E satisfying*
$$E \subseteq \mathcal{C} \subseteq \frac{2}{\gamma} \star E.$$

Proof. First, we show inductively that the convex body is contained in every constructed ellipsoid. By assumption, we have $\mathcal{C} \subseteq \bar{B}_n^{(2)}(c_0, R_{out}) = E(D_0, c_0)$. Now, we assume that $k \geq 0$ is an index such that $\mathcal{C} \subseteq E_k = E(D_k, c_k)$. The algorithm constructs the ellipsoid E_{k+1} only if there exists an element $x \in \mathcal{N}$ such that $c_k + \gamma Q_k^T x \notin \mathcal{C}$. Since $\mathcal{C}$ is convex, there exists an affine hyperplane that separates $c_k + \gamma Q_k^T x$ from $\mathcal{C}$. Such a hyperplane is given by the separation oracle, which provides a vector $a \in \mathbb{R}^n \backslash \{0\}$ such that $\langle a, c_k + \gamma Q_k^T x\rangle \geq \langle a, x\rangle$ for all $x \in \mathcal{C}$. Hence, the convex set $\mathcal{C}$ lies in the halfspace $\{x \in \mathbb{R}^n | \langle a, x\rangle \leq \langle a, c_k\rangle + \langle a, \gamma Q_k^T x\rangle\}$, that means

$$\mathcal{C} \subseteq E_k \cap \{x \in \mathbb{R}^n | \langle a, x\rangle \leq \langle a, c_k\rangle + \langle a, \gamma Q_k^T x\rangle\}. \tag{7.10}$$

Using the generalized Cauchy-Schwarz inequality, see Lemma 2.2.5 in Chapter 2, we obtain that

$$\begin{aligned}\langle a, \gamma Q_k^T x\rangle &\leq \gamma\sqrt{a^T D_k a} \cdot \sqrt{(Q_k^T x)^T D_k^{-1} (Q_k^T x)} \\ &= \gamma\sqrt{a^T D_k a} \cdot \sqrt{x^T Q_k (Q_k^T Q_k)^{-1} Q_k^T x} \\ &= \gamma\sqrt{a^T D_k a} \cdot \|x\|_2.\end{aligned}$$

Since $x \in \mathcal{N} \subseteq \mathbb{S}^{n-1}$ and $\gamma < 1/n$, we have

$$\langle a, \gamma Q_k^T x\rangle < \frac{1}{n}\sqrt{a^T D_k a}.$$

Hence, the affine hyperplane defined by the vector a provides a shallow cut of the ellipsoid E_k, see Definition 7.1.7. According to the conditions of Theorem 7.1.8 with parameter

$\zeta = \gamma$, the ellipsoid E_{k+1} is defined such that it contains the intersection $E_k \cap \{x \in \mathbb{R}^n | \langle a, x\rangle \leq \langle a, c_k\rangle + \zeta\sqrt{a^T Da}\}$ and it follows together with (7.10) that

$$\mathcal{C} \subseteq E_{k+1}.$$

The algorithm terminates after k iterations if all vectors $c_k + \gamma Q_k^T x$, $x \in \mathcal{N}$, are contained in the convex set $\mathcal{C}$. Since the set $\mathcal{N}$ is a 1-net of $\mathbb{S}^{n-1}$, it is guaranteed that the set $\mathcal{C}$ contains the ellipsoid E_k scaled by the factor $\gamma/2$,

$$\frac{\gamma}{2} \star E_k \subseteq \mathcal{C},$$

see Corollary 7.1.5. Altogether, the ellipsoid E_k satisfies

$$\frac{\gamma}{2} \star E_k \subseteq \mathcal{C} \subseteq E_k.$$

The algorithm outputs the symmetric positive definite matrix $D = (\gamma^2/4)D_k$ and the vector c_k. The ellipsoid defined by this matrix D and this vector c satisfies

$$E(D,c) = E\left(\frac{\gamma^2}{4}D_k, c_k\right) \subseteq \mathcal{C} \subseteq E(D_k, c_k) = \frac{2}{\gamma} \star E\left(\frac{\gamma^2}{4}D_k, c_k\right) = \frac{2}{\gamma} \star E(D,c),$$

which shows that the ellipsoid $E(D,c)$ is a $2/\gamma$-approximate Löwner-John ellipsoid of the convex set $\mathcal{C}$. □

It remains to show that the algorithm really terminates and outputs an ellipsoid. This can be guaranteed since in each iteration the volume of the constructed ellipsoid decreases by a single exponential factor. In the next lemma, we show that the volume of the ellipsoid which would be constructed in the N-th iteration is smaller than the volume of the convex set $\mathcal{C}$.

Lemma 7.1.14. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional bounded convex set with $\operatorname{vol}_n(\mathcal{C}) \geq r_{in}^n \operatorname{vol}_n(B_n^{(2)}(0,1))$ for some $r_{in} > 0$. If the ellipsoid $E_0 = \bar{B}_n^{(2)}(c_{out}, R_{out})$ defined in the initialization step of the rounding algorithm for convex sets, Algorithm 19, contains the convex set $\mathcal{C}$ and Step 2(b)ii of the algorithm is executed at least N-times, where*

$$N = 2\frac{(n+1)n}{(1-n\gamma)^2}(\log_2(R_{out}) - \log_2(r_{in})),$$

then

$$\operatorname{vol}_n(E_N) < \operatorname{vol}_n(\mathcal{C}).$$

Proof. According to Theorem 7.1.8 with $\zeta = \gamma$, in each iteration of step 2 in the algorithm, the volume of the constructed ellipsoid decreases by the factor

$$e^{-\frac{(1-n\gamma)^2}{2(n+1)}}.$$

Hence, the volume of the ellipsoid E_N is bounded by

$$\begin{aligned}\mathrm{vol}_n(E_N) &< \left(e^{-\frac{(1-n\gamma)^2}{2(n+1)}}\right)^N \cdot \mathrm{vol}_n(E_0)\\ &< 2^{-\frac{N(1-n\gamma)^2}{2(n+1)}} \cdot \mathrm{vol}_n(E_0).\end{aligned}$$

Since $E_0 = \bar{B}_n^{(2)}(c_{out}, R_{out})$, the volume of the initial ellipsoid is

$$\mathrm{vol}_n(E_0) = R_{out}^n \cdot \mathrm{vol}_n(B_n^{(2)}(0,1)) = 2^{n\log_2(R_{out})} \cdot \mathrm{vol}_n(B_n^{(2)}(0,1))$$

and we obtain by our definition of N that the volume of the ellipsoid E_N is smaller than the volume of $\mathcal{C}$,

$$\begin{aligned}\mathrm{vol}_n(E_N) &< 2^{-\frac{N(1-n\gamma)}{2(n+1)}+n\log_2(R_{out})} \cdot \mathrm{vol}_n(B_n^{(2)}(0,1))\\ &= 2^{n\cdot\log_2(r_{in})} \cdot \mathrm{vol}_n(B_n^{(2)}(0,1))\\ &\leq \mathrm{vol}_n(\mathcal{C}).\end{aligned}$$

□

Combining Lemma 7.1.13 and Lemma 7.1.14, we are able to prove the correctness of the algorithm.

Theorem 7.1.15. *Given a full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ by a separation oracle* $\mathrm{SEP}_{\mathcal{C}}$ *together with parameters $r_{in}, R_{out} > 0$ and a vector $c_{out} \in \mathbb{R}^n$ such that*

$$\mathcal{C} \subseteq \bar{B}_n^{(2)}(c_{out}, R_{out}) \text{ and } \mathrm{vol}(\mathcal{C}) \geq r_{in}^n \cdot \mathrm{vol}_n(B_n^{(2)}(0,1)),$$

and a parameter γ with $0 < \gamma < 1/n$, the rounding algorithm for convex sets, Algorithm 19, computes a $2/\gamma$-approximate Löwner-John ellipsoid, i.e., a positive definite matrix $D \in \mathbb{R}^{n\times n}$ and a vector $c \in \mathbb{R}^n$. The ellipsoid $E(D,c)$ satisfies

$$E(D,c) \subseteq \mathcal{C} \subseteq \frac{2}{\gamma} \star E(D,c).$$

If we assume that we are able to perform all computations exactly over $\mathbb{R}$, then the number of arithmetic operations and the number of calls to the oracle are at most

$$\frac{1}{(1-n\gamma)^2}(\log_2(R_{out}) - \log_2(r_{in}))2^{\mathcal{O}(n)}.$$

Proof. Each ellipsoid E, which is output by the algorithm satisfies $E \subseteq \mathcal{C} \subseteq (2/\gamma)\star E$, as we have seen in Lemma 7.1.13. Thus, we need to guarantee that the algorithm outputs something. Hence, we assume that the algorithm constructs all N ellipsoids, which is the only case, for which the algorithm does not output anything. The parameter N defined in the initialization step of the algorithm is at most

$$N = \lceil 2\frac{(n+1)n}{(1-n\gamma)^2}(\log_2(R_0) - \log_2(r_{in}))\rceil \geq 2\frac{(n+1)n}{(1-n\gamma)^2}(\log_2(R_0) - \log_2(r_{in})).$$

Using Lemma 7.1.14 together with the assumption that $\text{vol}_n(\mathcal{C}) \geq r_{in}^n \cdot \text{vol}_n(B_n^{(2)}(0,1))$, we see that after N iterations, the volume of the ellipsoid constructed in the N-th iteration is smaller than the volume of the convex body $\mathcal{C}$, $\text{vol}_n(E_N) < \text{vol}_n(\mathcal{C})$. This is a contradiction to the fact that each ellipsoid constructed by the algorithm contains the convex body as we have proven in Lemma 7.1.13.
Hence, there exists an iteration-step $k \leq N$ where the algorithm does not construct a new ellipsoid. In this step, the algorithm outputs an approximate Löwner-John ellipsoid.

The number of iteration-steps is at most $N \leq 2(n+1)n/(1-n\gamma)^2 \cdot (\log_2(R_0) - \log_2(r_{in})) + 1$. Since the set $\mathcal{N}$ can be constructed using at most $2^{\mathcal{O}(n)}$ arithmetic operations, it is easy to see that the number of arithmetic operations is at most

$$\frac{1}{(1-n\gamma)^2} \left(\log_2(R_{out}) - \log_2(r_{in})\right) 2^{\mathcal{O}(n)}.$$

□

It is easy to see that the rounding method runs in polynomial space if we can guarantee that the representation size of each constructed ellipsoid is polynomial in the dimension and in $\log_2(R_{out} \cdot r_{in}^{-1})$. Thus we need to take care of the size of the constructed ellipsoids. First, we prove that the coefficients of the matrices D_k and the vectors c_k do not become too large.

Lemma 7.1.16. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional bounded convex set, $n \geq 2$. Let $R_{out} > 0$ and $c_{out} \in \mathbb{R}^n$ such that $\mathcal{C} \subseteq \bar{B}_n^{(2)}(c_{out}, R_{out})$. If the parameter γ satisfies $-1 < \gamma < 1/n$, the rounding algorithm for convex sets, Algorithm 19, satisfies the following properties: For each ellipsoid $E_k = E(D_k, c_k)$, $k \geq 0$, constructed by the algorithm we have*

1. *$\|c_k\|_2 \leq R_{out} \cdot 2^k$,*
2. *$\|D_k\|_2 \leq R_{out}^2 \cdot 2^k$, and*
3. *$\|D_k^{-1}\|_2 \leq R_{out}^{-2} \cdot 9^k$,*

where $\|D_k\|_2$ denotes the spectral norm of the matrix D_k.

Proof. We will prove this by induction.
For $k = 0$, all statements are true since $c_0 = 0$ and $D_0 = R_{out}^2 I_n$, with $R_{out} \neq 0$. For $k > 0$, we start with the proof of the second statement. By definition, the spectral norm of the matrix D_{k+1} is

$$\|D_{k+1}\|_2 = \frac{n^2(1-\gamma^2)}{n^2-1} \left\| D_k - \frac{2}{n+1} \cdot \frac{1-n\gamma}{1-\gamma} \cdot \frac{D_k a (D_k a)^T}{a^T D_k a} \right\|_2$$

for some vector $a \in \mathbb{R}^n$.
It is easy to see that for symmetric positive definite matrices $A, B \in \mathbb{R}^{n \times n}$ we have

$\|A\|_2 \leq \|A+B\|_2$, see [MN99]. Since the matrices D_{k+1} and $-D_k a(D_k a)^T$ are symmetric positive definite, it follows that

$$\|D_{k+1}\|_2 \leq \frac{n^2(1-\gamma^2)}{n^2-1}\|D_k\|_2.$$

Now, using the induction hypothesis, we obtain the following upper bound for the spectral norm of D_{k+1},

$$\|D_{k+1}\|_2 \leq \frac{n^2(1-\gamma^2)}{n^2-1} R_{out}^2 \cdot 2^k \leq \frac{4}{3} R_{out}^2 \cdot 2^k < R_{out}^2 \cdot 2^{k+1},$$

where the second inequality is due to the fact that $1-\gamma^2 \leq 1$. To prove the third statement, we observe that the inverse of the matrix D_{k+1} is

$$D_{k+1}^{-1} = \frac{n^2-1}{n^2(1-\gamma^2)}\left(D_k^{-1} + \frac{2}{n-1} \cdot \frac{1+n\gamma}{1-\gamma}\frac{a \cdot a^T}{a^T D_k a}\right),$$

see Lemma 7.1.9. Using the triangle inequality, the spectral norm of this matrix is at most

$$\|D_{k+1}^{-1}\|_2 \leq \frac{n^2-1}{n^2(1-\gamma^2)}\left(\|D_k^{-1}\|_2 + \frac{2}{n-1} \cdot \frac{1+n\gamma}{1-\gamma} \cdot \frac{\|aa^T\|_2}{a^T D_k a}\right).$$

Since the spectral norm of the matrix aa^T is $a^T a$, we have

$$\frac{\|aa^T\|_2}{a^T D_k a} = \frac{a^T a}{a^T D_k a} \leq \max_{x \neq 0}\left|\frac{x^T x}{x^T D_k x}\right|,$$

which is exactly the spectral norm of D_k^{-1}, $\|D_k^{-1}\|_2$. Hence, we obtain that

$$\begin{aligned}\|D_{k+1}^{-1}\|_2 &\leq \frac{n^2-1}{n^2(1-\gamma^2)}\left(\|D_k^{-1}\|_2 + \frac{2}{n-1} \cdot \frac{1+n\gamma}{1-\gamma}\|D_k^{-1}\|_2\right) \\ &= \frac{n^2-1}{n^2(1-\gamma^2)}\left(1 + \frac{2}{n-1} \cdot \frac{1+n\gamma}{1-\gamma}\right)\|D_k^{-1}\|_2 \\ &= \frac{(n+1)^2}{n^2(1-\gamma)^2}\|D_k^{-1}\|_2.\end{aligned}$$

Now, it follows by the induction hypothesis that

$$\begin{aligned}\|D_{k+1}^{-1}\|_2 &\leq \frac{(n+1)^2}{n^2(1-\gamma)^2} \cdot R_{out}^{-2} \cdot 9^k \\ &\leq \frac{9}{4} \cdot 4 \cdot R_{out}^{-2} \cdot 9^k \\ &\leq 9 \cdot 9^k \cdot R_{out}^{-2} = 9^{k+1} \cdot R_{out}^{-2}.\end{aligned}$$

To prove the corresponding statement for the vector c_{k+1}, we consider the difference between this vector and the vector c_k. The Euclidean norm of this difference vector is at most

$$\begin{aligned}
\|c_{k+1} - c_k\|_2 &= \frac{1}{n+1}(1 - n\gamma)\frac{\|D_k a\|_2}{\sqrt{a^T D_k a}} \\
&= \frac{1}{n+1}(1 - n\gamma)\frac{\sqrt{a^T D_k^T D_k a}}{\sqrt{a^T D_k a}} \\
&= \frac{1}{n+1}(1 - n\gamma)\sqrt{\frac{a^T D_k^2 a}{a^T D_k a}} \\
&= \frac{1}{n+1}(1 - n\gamma)\sqrt{\frac{(D_k^{1/2} a)^T (D_k^{1/2})^T D_k^{1/2} (D_k^{1/2} a)}{(D_k^{1/2} a)^T (D_k^{1/2} a)}} \\
&\leq \frac{1}{n+1}(1 - n\gamma)\|D_k^{1/2}\|_2.
\end{aligned}$$

Since the eigenvalues of the square root of a positive definite matrix are the square root of the eigenvalues of the matrix, we are able to apply the induction hypothesis and we get

$$\|c_{k+1} - c_k\|_2 \leq \frac{1}{n+1}(1 - n\gamma)\sqrt{\|D_k\|_2} \leq \frac{1}{n+1}(1 - n\gamma) R_{out} 2^{k/2}.$$

Hence, we obtain the following upper bound for the norm of the vector c_{k+1}:

$$\begin{aligned}
\|c_{k+1}\|_2 &\leq \|c_{k+1} - c_k\|_2 + \|c_k\|_2 \\
&\leq \frac{1}{n+1}(1 - n\gamma) R_{out} \cdot 2^{k/2} + R_{out} \cdot 2^k \\
&= R_{out} \cdot \left(\frac{1}{n+1}(1 - n\gamma) 2^{k/2} + 2^k\right) \\
&= 2^k \cdot R_{out} \cdot \left(\frac{1}{n+1}(1 - n\gamma) 2^{k/2} + 1\right) \\
&\leq 2^{k+1} \cdot R_{out}.
\end{aligned}$$

□

Now, it follows directly that the coordinates of each ellipsoid constructed by the algorithm do not grow too fast.

Corollary 7.1.17. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a full-dimensional convex body. Let $R_{out} > 0$ and $c_{out} \in \mathbb{R}^n$ such that $\mathcal{C} \subseteq \bar{B}_n^{(2)}(c_{out}, R_{out})$. If the parameter γ satisfies $0 < \gamma < 1/n$, then each ellipsoid $E_k = E(D_k, c_k)$, $k \geq 0$, constructed by the rounding algorithm for convex sets, Algorithm 19, satisfies the following properties:*

- *Each coefficient d of the matrix D_k satisfies $|d| < R_{out}^2 \cdot 2^k$ and $|d| > R_{out}^2 9^{-k}$ if $d \neq 0$.*
- *Each coefficient c of the vector c_k satisfies $|c| < R_{out} \cdot 2^k$ and $|c| > (1 - n\gamma)/(n + 1)R \cdot 9^{-k}$ if $c \neq 0$.*

Obviously, this statement does not guarantee that the size of each instance constructed by the algorithm does not grow too fast. The other problem is that we are not able to compute the centers c_k of the ellipsoids E_k exactly, since we are not able to compute the square root over $\mathbb{Q}$. Thus, rounding is unavoidable. It is done as follows: We consider the binary representation of each coefficient and cut it after d digits behind the binary point for some parameter $d \in \mathbb{N}$. So we approximate each coefficient with a rational number whose denominator is at most 2^d. If we round the coefficients of the matrix D_k, we need to be careful, since we need to guarantee that we obtain a symmetric positive definite matrix.

For the constructed ellipsoids, rounding has the following effect on the algorithm: The rounding of the center c_k leads to a transformation of the ellipsoid, whereas rounding of the matrix D_k changes the shape of the ellipsoid. For the correctness of the algorithm, we need to guarantee that $\mathcal{C}$ is contained in the rounded ellipsoid which can be done using a careful scaling. In Chapter 3 of [GLS93] it is shown that it is enough to round to at most $8 \cdot N$ digits in the shallow cut ellipsoid method, where N is the number of iterations, and that the algorithm nevertheless outputs an approximate Löwner-John ellipsoid. Considering this together with Corollary 7.1.17 leads to the following upper bound on the size of the ellipsoids E_k,

$$\text{size}(E_k) \leq 2^N R_{out}^2,$$

where N is the number of iterations. Since

$$N \leq (1 - n\gamma)^{-2} \mathcal{O}(n^2) \left(\log_2(R_{out}) - \log_2(r_{in})\right),$$

the size of the ellipsoid E_k is upper bounded by

$$2^{(1-n\gamma)^{-2} \cdot \mathcal{O}(n^2)} R_{out}^3 \cdot r_{in}^{-1}.$$

In the following, we will ignore this difficulty and we will assume that all instances have polynomial encoding length and that all arithmetic operations can be carried out in polynomial time. We summarize this in the following.

Theorem 7.1.18. *Given a full-dimensional bounded convex set $\mathcal{C} \subseteq \mathbb{R}^n$ by a separation oracle together with parameters r_{in}, $R_{out} > 0$ and a vector $c_{out} \in \mathbb{R}^n$ such that*

$$\mathcal{C} \subset \bar{B}_n^{(2)}(c_{out}, R_{out}) \text{ and } \operatorname{vol}_n(\mathcal{C}) \geq r_{in}^n \operatorname{vol}_n(B_n^{(2)}(0,1)),$$

the rounding algorithm for convex sets, Algorithm 19, satisfies the following properties: The number of arithmetic operations of the algorithm and the number of calls to the oracle is at most

$$\frac{1}{(1 - n\gamma)^2} \left(\log_2(R_{out} \cdot r_{in}^{-1})\right)^{\mathcal{O}(1)} 2^{\mathcal{O}(n)}.$$

In each iteration, it computes an instance (D_k, c_k), where $D_k \in \mathbb{Q}^{n\times n}$ is a symmetric positive definite matrix and $c_k \in \mathbb{Q}^n$. The rounding algorithm runs in polynomial space and the size of each instance is at most

$$2^{\mathcal{O}(n^4)}(R_{out} \cdot r_{in}^{-1})^{\mathcal{O}(1)}.$$

The output of the algorithm is a $2/\gamma$-approximate Löwner-John ellipsoid, i.e., an ellipsoid $E \subseteq \mathbb{R}^n$ with $E \subseteq \mathcal{C} \subseteq 2/\gamma \star E$.

7.2. A rounding method for ℓ_p-bodies

In this section, we use the algorithmic framework presented in Section 7.1 to obtain an algorithm that computes an approximate Löwner-John ellipsoid for ℓ_p-bodies $B_{m,n}^{(p,V)}(t,\alpha)$ with $1 < p < \infty$, which we defined in Section 6.4.3 in Chapter 6. If the corresponding ℓ_p-body contains an integer vector, we can guarantee that the algorithm outputs an approximate Löwner-John ellipsoid. Otherwise, there are two possibilities: Either the algorithm outputs an approximate Löwner-John ellipsoid or it outputs that the ℓ_p-body does not contain an integer vector. We call this algorithm the rounding method for ℓ_p-bodies.

To apply the rounding method for bounded convex sets to the class of ℓ_p-bodies, we need to realize a separation oracle for ℓ_p-bodies. Given an ℓ_p-body $B_{m,n}^{(p,V)}(t,\alpha)$ together with a vector $y \in \mathbb{R}^m$ we need to decide whether y is contained in the ℓ_p-body. If this is not the case, we need to be able to compute an affine hyperplane that separates the vector $y \in \mathbb{R}^m$ from the ℓ_p-body $B_{m,n}^{(p,V)}(t,\alpha)$. Additionally, we need to determine parameters $R_{out}, r_{in} > 0$ and a vector $c_{out} \in \mathbb{R}^m$ such that

$$B_{m,n}^{(p,V)}(t,\alpha) \subseteq \bar{B}_m^{(2)}(c_{out}, R_{out}) \text{ and } \operatorname{vol}_m(B_{m,n}^{(p,V)}(t,\alpha)) \geq r_{in}^m \cdot \operatorname{vol}_m(B_m^{(2)}(0,1)).$$

The assumption that the ℓ_p-body contains an integer vector is only needed for the computation of a corresponding parameter r_{in}. In the next section, we will consider these aspects in detail, see Section 7.2.1. Then we will present a detailed description of the rounding algorithm for ℓ_p-bodies. This is done in Section 7.2.2.

7.2.1. Properties of ℓ_p-bodies

Computation of a circumscribed Euclidean ball

We have already seen in Lemma 6.3.3 in Chapter 6 that a general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ is contained in an n-dimensional Euclidean ball with radius $\alpha\sqrt{n}\|V\|_2$, where $\|V\|_2$ denotes the spectral norm of the matrix V. By intersecting this ball with the subspace $\bigcap_{i=m+1}^n H_{0,e_i}$ we can construct a circumscribed Euclidean ball for the ℓ_p-body $B_{m,n}^{(p,V)}(t,\alpha)$.

Lemma 7.2.1. *Let $B_{m,n}^{(p,V)}(t,\alpha)$ be an ℓ_p-body given by $V \in \mathbb{R}^{n\times n}$ nonsingular, $t \in \mathbb{R}^n$, $\alpha > 0$, and $1 < p < \infty$. Then $B_{m,n}^{(p,V)}(t,\alpha)$ is contained in an m-dimensional Euclidean ball with radius $\alpha\sqrt{n}\|V\|_2$. The center of this ball is given by the orthogonal projection of t onto* $\operatorname{span}(e_1,\dots,e_m)$.

Proof. The general ℓ_p-ball $B_n^{(p,V)}(t,\alpha)$ is contained in a Euclidean ball with radius $\alpha\sqrt{n}\|V\|_2$ centered at the vector t, see Lemma 6.3.3 in Chapter 6. Obviously, it follows that the ℓ_p-body $B_{m,n}^{(p,V)}(t,\alpha)$ is contained in the intersection of the Euclidean ball $\bar{B}_n^{(2)}(t,\alpha\sqrt{n}\|V\|_2)$ with the subspace $\bigcap_{i=m+1}^n H_{0,e_i}$, which is the m-dimensional Euclidean ball with radius $\alpha\sqrt{n}\|V\|_2$. The center of this ball is given by the orthogonal projection of t onto $\operatorname{span}(e_1,\dots,e_m)$. □

Next, we will prove a lower bound on the volume of an ℓ_p-body provided that it contains an integer vector.

Computation of a lower bound for the volume of an ℓ_p-body

The lower bound on the volume of an ℓ_p-body depends on the shape of the convex set, that means on the parameters defining it, and on the radius of a circumscribed Euclidean ball. For the proof of the lower bound, we consider a special representation of the ℓ_p-body. If we use that $\alpha = \alpha_n/\alpha_d$ with $\alpha_n, \alpha_d \in \mathbb{N}$ and consider the following convex function,

$$F : \mathbb{R}^m \to \mathbb{R},\ x \mapsto \alpha_d^p\|V^{-1}((x^T,0^{n-m})^T - t)\|_p^p - \alpha_n^p, \tag{7.11}$$

then $B_{m,n}^{(p,V)}(t,\alpha) = \{x \in \mathbb{R}^m | F(x) < 0\}$.

To illustrate the main idea of the proof which is due to Heinz [Hei05], we imagine that the function F is in addition differentiable and that we know an upper bound M on the length of its gradients $\nabla F(x)$, $x \in \mathbb{R}^m$, i. e., $\|\nabla F(x)\|_2 \le M$ for all $x \in \mathbb{R}^m$. Furthermore, we assume that we know some parameter $\epsilon > 0$ such that there exists a vector $\hat{x} \in \mathbb{R}^m$ with $F(\hat{x}) \le -\epsilon < 0$.

Since for every convex function the first-order Taylor approximation is a global underestimator of the function (first-order convexity condition), see Lemma 2.1.8 in Chapter 2, we obtain for all $x \in \mathbb{R}^m$ that

$$F(\hat{x}) \ge F(x) + \nabla F(x)^T(\hat{x} - x).$$

Using the Cauchy-Schwarz inequality, this yields the upper bound

$$F(x) \le F(\hat{x}) + \nabla F(x)^T(x - \hat{x}) \le -\epsilon + M\|x - \hat{x}\|_2.$$

Hence, if a vector $x \in \mathbb{R}^m$ satisfies $\|x - \hat{x}\|_2 \le \epsilon/M$, then $F(x) < 0$ and it is contained in the set $B_{m,n}^{(p,V)}(t,\alpha)$. This shows that $B_{m,n}^{(p,V)}(t,\alpha)$ contains a Euclidean ball with radius ϵ/M centered around $\hat{x}$ and that the volume of $B_{m,n}^{(p,V)}(t,\alpha)$ is at least $(\epsilon/M)^m \operatorname{vol}_m(B_m^{(2)}(0,1))$.

For the function F defined in (7.11) we can compute such a parameter ϵ since we can show that F is enumerable. Revisiting Definition 4.3.14 in Chapter 4 we see that a function $f : \mathbb{R}^m \to \mathbb{R}$ is enumerable if there exists an integer $K \in \mathbb{N}$ such that $K \cdot F(x) \in \mathbb{Z}$ for all $x \in \mathbb{Z}^m$. So, $F(x)$ is a rational number with denominator at most K for every integer vector $x \in \mathbb{Z}^n$. Hence, if $B_{m,n}^{(p,V)}(t,\alpha)$ contains an integer vector $\hat{x} \in \mathbb{Z}^m$, then $F(\hat{x}) \leq -1/K < 0$. In the following claim, we give an upper bound on the number K.

Claim 7.2.2. *Let $F : \mathbb{R}^m \to \mathbb{R}$ be a function defined as in (7.11) given by a nonsingular matrix $V \in \mathbb{Q}^{n\times n}$, a vector $t \in \mathbb{Q}^n$ and $\alpha_n, \alpha_d \in \mathbb{N}$. Let S be an upper bound on the size of V^{-1}, t, α_n and α_d.*
Then, there exists an integer $K \leq S^{2n^2p}$ such that $K \cdot F(x) \in \mathbb{Z}$ for all $x \in \mathbb{Z}^m$.

Proof. Since $\alpha_n, \alpha_d \in \mathbb{N}$, we observe that $F(x) \in \mathbb{Z}$ if all coefficients of the matrix V^{-1} and the vector t are integers. If $V^{-1} = (v_{ij}) \in \mathbb{Q}^{n\times n}$ and $t = (t_i) \in \mathbb{Q}^n$ the coefficients of the vector $V^{-1}t$ are rationals of the form $\sum_{j=1}^n v_{ij}t_j$. That means, each coefficient is the sum of n rational numbers whose denominators are at most S^2.
Hence, the multiplication of this vector with the product of these denominators yields an integer vector. The multiplication of V^{-1} with the same number yields an integer matrix. Hence, there exists a number which is at most $(S^2)^{n^2} = S^{2n^2}$ such that $V^{-1}((x^T, 0^{n-m})^T - t)$ becomes an integer if multiplied with this number. Since F consists of the p-th power of an ℓ_p-norm, there exists a number which is at most $(S^{2n^2})^p = S^{2n^2p}$ such that $F(x)$ becomes an integer if multiplied with this number. □

Now, the main remaining problem is that the function F is not differentiable. Hence, we need to modify the idea described above and work with the subgradient instead of the gradient. We start with a short overview about subgradients.

Definition 7.2.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and $x \in \mathbb{R}^n$. A vector $g \in \mathbb{R}^n$ is called a* subgradient of f at x *if the following holds,*

$$f(z) \geq f(x) + \langle g, z - x\rangle \text{ for all } z \in \mathbb{R}^n. \tag{7.12}$$

The inequality (7.12) is called *subgradient inequality*. Geometrically, this inequality means that the graph of the affine function $z \mapsto f(x)+\langle g, z-x\rangle$ is a supporting hyperplane of the epigraph of f at $(x, f(x))$ as it is shown in Figure 7.6. The subgradient inequality is the corresponding equivalent to the first-order convexity condition for differentiable convex functions. If f is differentiable, then the subgradient is unique and it is simply the gradient of f at x. For a more detailed introduction into subgradients see [Roc70] and [Pol87].

Now we can prove a lower bound on the volume of the set $B_{m,n}^{(p,V)}(t,\alpha)$ under the assumption that for all $R > 0$ and $y \in \bar{B}_m^{(2)}(0,R)$ the length of a corresponding subgradient is bounded.

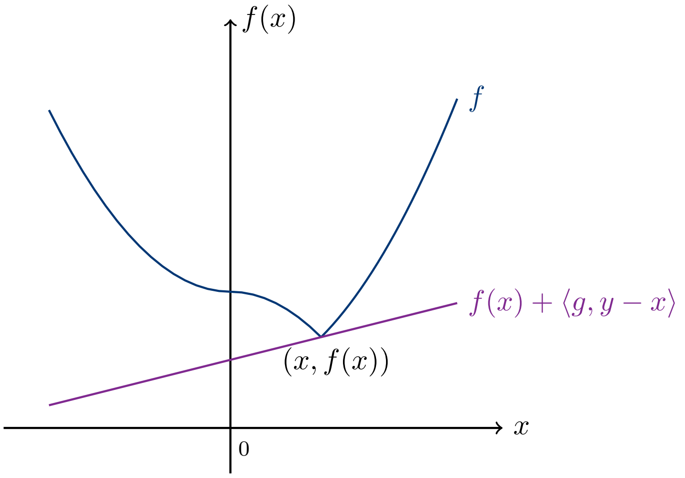

Figure 7.6.: **Subgradient of a convex function.** The subgradient g defines a supporting hyperplane of the epigraph of the function f at the point $(x, f(x))$.

Lemma 7.2.4. *Let $B_{m,n}^{(p,V)}(t,\alpha)$ be an ℓ_p-body given by $V \in \mathbb{Q}^{n\times n}$ nonsingular, $t \in \mathbb{Q}^n$, $\alpha = \alpha_n/\alpha_d > 0$ and $1 < p < \infty$. Let $F : \mathbb{R}^m \to \mathbb{R}$ be a function defined as in (7.11). Let S be an upper bound on the size of $B_{m,n}^{(p,V)}(t,\alpha)$. Let $R > 0$ such that $B_{m,n}^{(p,V)}(t,\alpha)$ is contained in a Euclidean ball with radius R centered at the origin. Assume that there exists $M \in \mathbb{R}$ such that the following holds: For all $y \in \bar{B}_m^{(2)}(0,R)$ there exists a subgradient $g \in \mathbb{R}^m$ of F at y which satisfies $\|g\|_2 \leq M$. If $B_{m,n}^{(p,V)}(t,\alpha)$ contains an integer vector $\hat{x} \in \mathbb{Z}^m$, then*

$$\mathrm{vol}_m(B_{m,n}^{(p,V)}(t,\alpha)) > (S^{2n^2p}M)^{-m} \cdot \mathrm{vol}_m(B_m^{(2)}(0,1)).$$

Proof. Let $g \in \mathbb{R}^m$ be a subgradient of F at the vector $y \in \bar{B}_m^{(2)}(0,R)$ which satisfies $\|g\|_2 \leq M$. Then it follows from the subgradient inequality (7.12) for $\hat{x} \in \mathbb{Z}^m$ that

$$F(\hat{x}) \geq F(y) + \langle g, \hat{x} - y\rangle.$$

As we have seen in Claim 7.2.2, $F(\hat{x})$ is a rational number with denominator at most S^{2n^2p}. Since $F(\hat{x}) < 0$ and using the Cauchy-Schwarz inequality, we obtain

$$F(y) \leq F(\hat{x}) + \langle g, y - \hat{x}\rangle \leq -S^{-2n^2p} + \|g\|_2 \cdot \|y-\hat{x}\|_2 \leq -S^{-2n^2p} + M\|y-\hat{x}\|_2$$

which shows that every vector $y \in \bar{B}_m^{(2)}(0,R)$ with $\|y-\hat{x}\|_2 \leq S^{-2n^2p}/M$ satisfies $F(y) < 0$ and is contained in $B_{m,n}^{(p,V)}(t,\alpha)$.

Hence, the ℓ_p-body $B_{m,n}^{(p,V)}(t,\alpha)$ contains a ball with radius $(S^{2n^2p}M)^{-1}$ centered at $\hat{x}$ and the claimed lower bound for the volume follows directly. □

This result shows that we need to compute for every vector $y \in \bar{B}_m^{(2)}(0, R)$ an upper bound on the length of a corresponding subgradient of F depends only on the parameter R if we want to obtain a lower bound on the volume of the ℓ_p-body $B_{m,n}^{(p,V)}(t, \alpha)$. To do this, we develop an explicit expression of a subgradient of F in the following. We start with the computation of a subgradient of the following simple function.

Lemma 7.2.5. *Let $y \in \mathbb{R}^n$ and $1 < p < \infty$. Then a subgradient $g \in \mathbb{R}^n$ of the function*

$$F_p : \mathbb{R}^n \to \mathbb{R}, \ x \mapsto \sum_{i=1}^{n} |x_i|^p$$

at the vector y is given by $g = (g_1, \ldots, g_n)^T$, where

$$g_i := \operatorname{sign}(y_i) \cdot |y_i|^{p-1}.$$

Proof. The proof consists of showing that the vector g satisfies the subgradient inequality (7.12). Since F_p is a nonnegative combination of the functions $x \mapsto |x_i|^p$, $1 \leq i \leq n$, it is enough to consider the case where $n = 1$.
For all $z \in \mathbb{R}$ and $0 < \lambda \leq 1$ it follows from the convexity of the function F_p that

$$F_p(y + \lambda(z - y)) \leq (1 - \lambda)F_p(y) + \lambda F_p(z)$$

or equivalently that

$$F_p(z) \geq \frac{1}{\lambda} \left(F_p(y + \lambda(z - y)) - (1 - \lambda)F_p(y)\right) = F_p(y) + \frac{1}{\lambda} \left(F_p(y + \lambda(z - y)) - F_p(y)\right).$$

Hence, the vector $g \in \mathbb{R}$ satisfies $F_p(z) \geq F_p(y) + g \cdot (z - y)$ if we can show that

$$F_p(y + \lambda(z - y)) - F_p(y) \geq \lambda \cdot g \cdot (z - y) = \lambda \operatorname{sign}(y) \cdot |y|^{p-1}(z - y).$$

By definition of F_p, we have $F_p(y + \lambda(z - y)) - F_p(y) = |y + \lambda(z - y)|^p - |y|^p$. Since for all $a, b \in \mathbb{R}$, $m \in \mathbb{N}$, it holds that $b^m - a^m = (b - a) \cdot \sum_{i=0}^{m-1} b^{m-1-i}a^i$, we see that

$$\begin{aligned} |y + \lambda(z - y)|^p - |y|^p &= (|y + \lambda(z - y)| - |y|) \cdot \sum_{i=0}^{p-1} |y + \lambda(z - y)|^{p-1-i} \cdot |y|^i \\ &\geq (|y + \lambda(z - y)| - |y|)\, |y|^{p-1}. \end{aligned}$$

Since for all $a, b \in \mathbb{R}$, $|a| - |b| \geq \operatorname{sign}(b) \cdot (a - b)$, this is at least $\lambda \cdot \operatorname{sign}(y)(z - y) \cdot |y|^{p-1}$. □

To compute a subgradient of the function F defined as in (7.11), we combine this result with the following lemma, which shows how a subgradient changes if we consider an affine transformation of the variables or the function.

Lemma 7.2.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function.*

- *Let $h_1 : \mathbb{R}^n \to \mathbb{R}$ be defined by $h_1(x) := f(Ax + \beta)$, where $A \in \mathbb{R}^{n \times n}$ is a nonsingular matrix and $\beta \in \mathbb{R}^n$. Let $g_1 \in \mathbb{R}^n$ be a subgradient of f at the vector $Ay + \beta$. Then, the vector $A^T g_1$ is a subgradient of h_1 at the vector y.*

- *Let $h_2 : \mathbb{R}^n \to \mathbb{R}$ be defined by $h_2(x) := a \cdot f(x) + b$, where $a \in \mathbb{R}\backslash\{0\}$ and $b \in \mathbb{R}$. Let $g_2 \in \mathbb{R}^n$ be a subgradient of f at the vector $y \in \mathbb{R}^n$. Then $a \cdot g_2$ is a subgradient of h_2 at the vector y.*

Proof. Let $g_1 \in \mathbb{R}^n$ be a subgradient of f at the vector $Ay + \beta$, i.e., $f(z) \geq f(Ay + \beta) + \langle g, z - (Ay + \beta)\rangle$ for all $z \in \mathbb{R}^n$. Thus, for all $z \in \mathbb{R}^n$ it holds that

$$\begin{aligned} h_1(z) &= f(Az + \beta) \\ &\geq f(Ay + \beta) + \langle g_1, Az + \beta - (Ay + \beta)\rangle \\ &= f(Ay + \beta) + \langle g_1, A(z - y)\rangle \\ &= h(y) + \langle A^T g_1, z - y\rangle, \end{aligned}$$

which shows that $A^T g_1 \in \mathbb{R}^n$ is a subgradient of h_1 at the vector y.
Let $g_2 \in \mathbb{R}^n$ be a subgradient of f at the vector $y \in \mathbb{R}^n$, i.e., $f(z) \geq f(y) + \langle g_2, z - y\rangle$ for all $z \in \mathbb{R}^n$. Thus, for all $z \in \mathbb{R}^n$ it holds that

$$\begin{aligned} h_2(z) &= a \cdot f(z) + b \\ &\geq a(f(y) + \langle g_2, z - y\rangle) + b \\ &= af(y) + b + \langle a \cdot g_2, z - y\rangle \\ &= h_2(y) + \langle a \cdot g_2, z - y\rangle. \end{aligned}$$

□

If we apply this result with $A = V^{-1}$, $\beta = -V^{-1}t$ and $a = \alpha_d^p$, $b = \alpha_n^p$, and restrict the subgradient to its first m coordinates we are able to give an explicit expression of a subgradient of the function F.

Lemma 7.2.7. *For $m, n \in \mathbb{N}$, $m \leq n$, a subgradient at the vector $y \in \mathbb{R}^m$ of the function*

$$F : \mathbb{R}^m \to \mathbb{R},\ x \mapsto \alpha_d^p \|V^{-1}((x^T, 0^{n-m})^T - t)\|_p^p - \alpha_n^p,$$

where $V \in \mathbb{R}^{n\times n}$ is nonsingular, $t \in \mathbb{R}^n$, $\alpha_n, \alpha_d \in \mathbb{N}$ and $1 < p < \infty$, is given by the vector $\alpha_d^p g \in \mathbb{R}^m$ defined by

$$g = ((V^{-1})^T \bar{g})_{\{1,\dots,m\}},$$

where $\bar{g} \in \mathbb{R}^n$ is defined by

$$\bar{g}_i = \operatorname{sign}([V^{-1}(y - t)]_i) \cdot |[V^{-1}(y - t)]_i|^p.$$

Using this explicit expression of the subgradient, we are able to give an upper bound on its length. In the following we denote by $x_{\{1,\dots,m\}} \in \mathbb{R}^m$ the vector in $\mathbb{R}^m$ which consists of the first m coordinates of the vector $x \in \mathbb{R}^n$.

Lemma 7.2.8. *Let $y \in \bar{B}_m^{(2)}(0,R) \subseteq \mathbb{R}^m$. Let $V \in \mathbb{Q}^{n\times n}$ be nonsingular, $t \in \mathbb{Q}^n$, α_n, $\alpha_d \in \mathbb{N}$, $1 < p < \infty$. Let $g \in \mathbb{R}^m$ defined by $g = [(V^{-1})^T \bar{g}]_{\{1,\ldots,m\}}$ where $\bar{g} \in \mathbb{R}^n$ is given as $\bar{g}_i := \operatorname{sign}([V^{-1}(y-t)]_i)|[V^{-1}(y-t)]_i|^p$, $1 \le i \le n$. Then*

$$\|\alpha_d^p g\|_2 \le m \cdot \left(\alpha_d n S^2 R\right)^{p+1}$$

where S is an upper bound on the size of V^{-1} and t.

Proof. Since $\|g\|_2 \le m \cdot \max\{|g_i| | 1 \le i \le m\}$, it is enough to compute an upper bound on the coefficient of the vector g.
If $V^{-1} = (v_{ij})_{i,j} \in \mathbb{Q}^{n\times n}$ and $t = (t_i)_i \in \mathbb{Q}^n$, the k-th coefficient, $1 \le k \le n$, of the vector $V^{-1}(y-t)$ is given by

$$|[V^{-1}(y-t)]_k| \le \sum_{j=1}^{n} |v_{kj} \cdot (y_j - t_j)|.$$

Since the coefficients of V^{-1} and t are at most S and since each coefficient of y is at most R (in absolute values), we obtain

$$|[V^{-1}(y-t)]_k| \le n \cdot S(R+S) \le nRS^2.$$

Hence, each coefficient of the vector $\bar{g}$ is at most

$$|g_i| \le (nRS^2)^p.$$

With the same argumentation, we obtain that each coefficient of the vector g is at most

$$|g_i| \le n \cdot S(nRS^2)^p \le (nS^2R)^{p+1}.$$

□

Using this upper bound together with Lemma 7.2.4 and the upper bound of a radius of a circumscribed Euclidean ball, we get the following lower bound on the volume of $B_{m,n}^{(p,V)}(t,\alpha)$.

Lemma 7.2.9. *Let $B_{m,n}^{(p,V)}(t,\alpha)$ be an ℓ_p-body where $t \in \mathbb{R}^n$, $V \in \mathbb{Q}^{n\times n}$ is nonsingular, $\alpha \in \mathbb{Q}^+$ and $1 < p < \infty$.*
If $B_{m,n}^{(p,V)}(t,\alpha)$ contains an integral vector, its volume is at least

$$\operatorname{vol}_m(B_{m,n}^{(p,V)}(t,\alpha)) \ge \left(S^{2(n^2+2)} m^2 n^2 \|V\|_2\right)^{-m(p+1)} \cdot \operatorname{vol}_m(B_m^{(2)}(0,1)),$$

where S is an upper bound on the size of V^{-1} and t.

Proof. It follows from Lemma 7.2.1 that the convex body $B_{m,n}^{(p,V)}(t,\alpha)$ is contained in a Euclidean ball centered at the origin, whose radius is at most $\alpha\sqrt{n}\|V\|_2 + mS$. Hence, if we choose $R := \alpha\sqrt{n}m\|V\|_2 \cdot S$, the Euclidean ball $B_m^{(2)}(0,R)$ contains $B_{m,n}^{(p,V)}(t,\alpha)$.

Combining this with the result from Lemma 7.2.8, we obtain from Lemma 7.2.4 that the volume of $B_{m,n}^{(p,V)}(t,\alpha)$ is at least the volume of the Euclidean unit ball $B_m^{(2)}(0,1)$ multiplied with the factor

$$\left(S^{2n^2p} \cdot m(\alpha_d n S^2 \alpha \sqrt{n} m \|V\|_2 \cdot S)^{p+1}\right)^{-m} \geq \left(S^{2n^2} mnS^3 \alpha_d \alpha \sqrt{n} m \|V\|_2\right)^{-m(p+1)}.$$

Since $\alpha_d \cdot \alpha = \alpha_n \leq S$, the statement follows. □

Realization of a separation oracle

Now we show that we are able to realize a separation oracle for ℓ_p-bodies. Again, we use here that $B_{m,n}^{(p,V)}(t,\alpha)$ can be characterized as $\{x \in \mathbb{R}^m | F(x) < 0\}$ if the function F is defined as in (7.11). Since we are able to compute a subgradient of this function efficiently, we are able to compute a separating hyperplane efficiently.

Lemma 7.2.10. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and $\mathcal{C}_\alpha := \{x \in \mathbb{R}^n | f(x) < \alpha\}$ be the corresponding convex set for some $\alpha > 0$. Let $y \in \mathbb{R}^n$ with $y \notin \mathcal{C}_\alpha$. Then, any subgradient $g \in \mathbb{R}^n$ of f at y defines a hyperplane that strictly separates y from $\mathcal{C}_\alpha$, i.e., $\langle g, x\rangle \leq \langle g, y\rangle$ for all $x \in \mathcal{C}_\alpha$.*

The proof of this lemma follows directly from the subgradient inequality (7.12).

Proof. Let $g \in \mathbb{R}^n$ be a subgradient of f at y. Then for all $x \in \mathbb{R}^n$ we have

$$f(x) \geq f(y) + \langle g, x - y\rangle$$

or equivalently

$$\langle g, x\rangle \leq f(x) - f(y) + \langle g, y\rangle.$$

If $x \in \mathcal{C}_\alpha$ we have $f(x) < \alpha$ and since $y \notin \mathcal{C}_\alpha$, we have $f(y) > \alpha$. Hence, $f(x) - f(y) < 0$. □

Thus, Lemma 7.2.7 leads to an efficient realization of a separation oracle for an ℓ_p-body.
Together with the results from Lemma 7.2.1 and Lemma 7.2.9, this shows how we can realize the general rounding method for convex sets presented in Section 7.1 to obtain an algorithm that computes an approximate Löwner-John ellipsoid for ℓ_p-bodies.

7.2.2. Description and analysis of the algorithm

Using the results from Section 7.2.1, we are able to present a concrete realization of the rounding method for bounded convex sets presented in Section 7.1.3 that computes approximate Löwner-John ellipsoids for ℓ_p-bodies $B_{m,n}^{(p,V)}(t,\alpha)$ which contain an integer vector. Essentially, the rounding algorithm for ℓ_p-bodies is a strict realization of the rounding algorithm for bounded convex sets presented in Algorithm 19.

The input of the algorithm is an ℓ_p-body $B_{m,n}^{(p,V)}(t,\alpha) \subseteq \mathbb{R}^m$ given by a nonsingular matrix $V \in \mathbb{Q}^{n\times n}$, a vector $t \in \mathbb{Q}^n$ and parameters $\alpha > 0$ and $1 < p < \infty$. In the initialization, the algorithm computes a vector $c_0 \in \mathbb{R}^m$ and a parameter $R_{out} > 0$ such that $B_{m,n}^{(p,V)}(t,\alpha) \subseteq \bar{B}_m^{(2)}(c_{out}, R_{out})$. This is done according to Lemma 7.2.1. The parameter R_{out} together with a parameter r_{in}, which is determined according to Lemma 7.2.9, are used to determine an upper bound for the number of iterations.
After the initialization, the algorithm continues iteratively in the same way as the rounding method for bounded convex sets. The only difference is in step 3(b)ii), where the algorithm computes a separating hyperplane directly according to Lemma 7.2.10 instead of using a separation oracle. A detailed description of the algorithm is presented in Algorithm 20.

The correctness of the algorithm follows directly from the previous statements.

Theorem 7.2.11. *(Theorem 6.4.14 restated.)*
Let $B_{m,n}^{(p,V)}(t,\alpha) \subseteq \mathbb{R}^m$ be an ℓ_p-body given by $V \in \mathbb{Q}^{n\times n}$ nonsingular, $t \in \mathbb{Q}^n$, $\alpha > 0$ and $1 < p < \infty$. Given such a convex set together with a parameter γ with $0 < \gamma < 1/m$, the rounding method for ℓ_p-bodies, Algorithm 20, satisfies the following properties:

- *The output of the algorithm is one of the following:*
  - *Either it outputs that $B_{m,n}^{(p,V)}(t,\alpha)$ does not contain an integer vector, or*
  - *it outputs a $2/\gamma$-approximate Löwner-John ellipsoid, i.e., a positive definite matrix $D \in \mathbb{Q}^{m\times m}$ and a vector $c \in \mathbb{Q}^m$ defining the ellipsoid $E(D,c)$ such that*

    $$E(D,c) \subseteq B_{m,n}^{(p,V)}(t,\alpha) \subseteq \frac{2}{\gamma} \star E(D,c).$$

    In this case, the size of the ellipsoid is at most $2^{\mathcal{O}(n^4)} r^{\mathcal{O}(n^2 p)}$.
- *The algorithm runs in polynomial space and the number of arithmetic operations is at most*

  $$\frac{p}{(1-m\gamma)^2}(n \cdot \log_2(r))^{\mathcal{O}(1)} 2^{\mathcal{O}(m)}.$$

Here, r is an upper bound on the size of the ℓ_p-body.

Proof. The correctness of the algorithm follows directly from Theorem 7.1.15 in combination with Lemma 7.2.1 and Lemma 7.2.9.

The number of arithmetic operations of the rounding method is mainly determined by the size of the set $\mathcal{N}$ and the number of iterations. The number of iterations is mainly influenced by the radius R_{out} of the circumscribed Euclidean ball and the lower bound on the volume of the polytope r_{in}. The circumscribed Euclidean ball has radius

Algorithm 20 Rounding method for ℓ_p-bodies

```
Input:
  • An $\ell_p$-body $B^{(p,V)}_{m,n}(t,\alpha)$ given by $V \in \mathbb{Q}^{n \times n}$ nonsingular, a vector $t \in \mathbb{Q}^n$ and a
    parameter $\alpha \in \mathbb{Q}$, $\alpha = \alpha_n/\alpha_d > 0$, and
  • a parameter $\gamma$ with $0 < \gamma < 1/m$.
Output: An ellipsoid $E \subseteq \mathbb{R}^n$ given by a symmetric positive definite matrix $D$ and a
  center $c$, or the statement that $B^{(p,V)}_{m,n}(t,\alpha)$ does not contain an integer vector.
  1. Set
     a) $r \leftarrow \max\{\text{size}(V^{-1}), \text{size}(t)\}$,
     b) $R_{out} \leftarrow \alpha\sqrt{n}\|V\|_2$, and
     c) $r_{in} \leftarrow (r^{2(n^2+2)} m^2 n^2 \|V\|_2)^{-m(p+1)}$.
  2. Set
     a) $N \leftarrow \lceil 2\frac{(m+1)m}{(1-m\gamma)^2}(\log_2(R_{out}) - \log_2(r_{in}))\rceil$,
     b) $D_0 \leftarrow R^2_{out} \cdot I_m$, and $c_0 \leftarrow \sum_{i=1}^m \langle t, e_i\rangle e_i$.
  3. For $0 \le k \le N$,
     a) compute a decomposition of the matrix $D_k$, $D_k = Q_k^T Q_k$.
     b) Check if there exists $x \in \{x/\|x\|_2 |\ x \in \mathbb{Z}^m \cap \bar{B}^{(2)}_m(0, 2\sqrt{m})\backslash\{0\}\}$ such that
           $$\tilde{c}_k \leftarrow c_k + \gamma Q_k^T x \notin B^{(p,V)}_{m,n}(t,\alpha).$$
        i. If no such element exists, output $E((\gamma^2/4) \cdot D_k, c_k)$.
        ii. Otherwise, compute
              $$a \leftarrow ((V^{-1})^T g)_{\{1,\dots,m\}} \in \mathbb{R}^m,$$
            where $g_i \leftarrow \text{sign}([V^{-1}((0^{n-m}, \tilde{c}_k^T)^T - t)]_i) \cdot \big|[V^{-1}((0^{n-m}, \tilde{c}_k^T)^T - t)]_i\big|^p$
            for $1 \le i \le n$ and set
              $$c_{k+1} \leftarrow c_k - \frac{1 - m\gamma}{m+1} \cdot \frac{D_k a}{\sqrt{a^T D_k a}} \text{ and}$$
              $$D_{k+1} \leftarrow \frac{m^2(1-\gamma^2)}{m^2-1} \left( D_k - \frac{2}{m+1} \cdot \frac{1-m\gamma}{1-\gamma} \cdot \frac{D_k a (D_k a)^T}{a^T D_k a} \right).$$
  4. Output that $B^{(p,V)}_{m,n}(t,\alpha)$ does not contain an integer vector.
```

$R_{out} = \alpha\sqrt{n}\|V\|_2$. Since the spectral norm of a matrix is smaller than $\|V\|_2 \leq n \cdot \max\{|v_{i,j}||1 \leq i,j \leq n\}$ and we obtain

$$R_{out} \leq \alpha n^{3/2}\,\text{size}(V).$$

The size of the matrix V is at most

$$n^{n/2}\,\text{size}(V^{-1})^{n(n-1)} \leq n^{n/2}r^{n(n-1)},$$

since r is an upper bound on the size of V^{-1}. Since r is also an upper bound on the parameter α, we obtain

$$R_{out} \leq n^{(n+3)/2}r^{n(n-1)+1} \leq r^{4n^2}. \tag{7.13}$$

The lower bound on the volume of the ℓ_p-body is given by the parameter

$$r_{in}^{-1} = \left(r^{2(n^2+2)}m^2n^2\|V\|_2\right)^{(p+1)m}.$$

With the same argumentation as above, we obtain

$$\begin{aligned} r_{in}^{-1} &\leq \left(r^{2(n^2+2)}m^2n^2n^{n/2}\cdot r^{n(n-1)}\right)^{(p+1)m} \qquad (7.14)\\ &\leq \left(r^{2(n^2+2)+4+n+n^2}\right)^{(p+1)m}\\ &\leq \left(r^{10n^2}\right)^{(p+1)m}. \end{aligned}$$

This shows that the number of iterations is at most

$$\begin{aligned} N &\leq 2\frac{m(m+1)}{(1-m\gamma)^2}\log_2(R_{out}\cdot r_{in}^{-1}) + 1\\ &\leq 2\frac{m(m+1)}{(1-m\gamma)^2}\cdot\log_2(r^{4n^2}\cdot r^{10n^2m(p+1)}) + 1\\ &\leq 2\frac{m(m+1)}{(1-m\gamma)^2}\cdot\log_2(r^{11n^2m(p+1)}) + 1\\ &\leq (p+1)\frac{n^{\mathcal{O}(1)}}{(1-m\gamma)^2}\log_2(r). \end{aligned}$$

In each iteration we need to check for each element x from the set $\mathcal{N}$, whether the vector $c_k + \gamma Q_k^T x\ \gamma \cdot x$ is contained in the ℓ_p-body $B_{m,n}^{(p,V)}(t,\alpha)$. Since $|\mathcal{N}| \leq 2^{4m}$, as we have shown in Lemma 7.1.6, this can be done using $n^{\mathcal{O}(1)}2^{\mathcal{O}(m)}$ arithmetic operations. The other operations are standard matrix operations. Hence, we obtain the following upper bound for the number of arithmetic operations used by the algorithm

$$\frac{p+1}{(1-m\gamma)^2}(n\cdot\log_2(r))^{\mathcal{O}(1)}2^{\mathcal{O}(m)}.$$

According to Theorem 7.1.18 we can assume that the algorithm runs in polynomial space and that the size of each computed instance E_k is at most $2^{\mathcal{O}(n^4)}(R_{out} \cdot r_{in}^{-1})^{\mathcal{O}(1)}$. Using (7.13) and (7.14), we obtain

$$2^{\mathcal{O}(n^4)} \left(r^{4n^2} \cdot r^{10n^2 m(p+1)} \right)^{\mathcal{O}(1)} = 2^{\mathcal{O}(n^4)} r^{\mathcal{O}(n^3 p)}.$$

□

7.3. A rounding method for polytopes

The general rounding method for bounded convex sets can also be used to compute a $2/\gamma$-approximate Löwner-John ellipsoid for full-dimensional polytopes for some parameter $0 < \gamma < 1/n$. As polytopes can be characterized as the intersection of finitely many halfspaces, we are even able to improve the general rounding method in this special case. This leads to an algorithm originally developed from Goffin and Lenstra, see [Gof84], [Len83]. In this section, we will describe this algorithm that computes a $1/\gamma$-approximate Löwner-John ellipsoid and whose number of arithmetic operations is polynomial in the dimension, in the number of constraints and logarithmic in the size of the polytope. In contrast to the class of ℓ_p-bodies, we can guarantee that the algorithm computes an approximate Löwner-John ellipsoid.

We observe that it is important that the number of arithmetic operations is polynomial in the number of constraints defining the polytope. For example, if we consider the unit ball of the ℓ_1-norm, then this ball can be described as the polytope $\{x \in \mathbb{R}^n | \langle x, e \rangle \leq 1 \text{ and } \langle x, e \rangle \geq -1 \text{ for all } e \in \{1, -1\}^n\}$ using 2^{n+1} constraints. Hence, in this case we obtain an algorithm which is single exponential in the dimension.

Before we describe how we modify the rounding method, we first describe how we realize a separation oracle for polytopes. Furthermore we show how for a given polytope we can compute a circumscribed Euclidean ball and a lower bound on its volume.

7.3.1. Properties of polytopes

In the following we always assume that we are given a full-dimensional polytope $P \subseteq \mathbb{R}^n$ given by a set of integral constraints $a_i \in \mathbb{R}^n$, $1 \leq i \leq s$, together with a set of parameters $\{\beta_1, \dots, \beta_s\} \subseteq \mathbb{N}$,

$$P = \{x \in \mathbb{R}^n | \langle a_i, x \rangle \leq \beta_i \text{ for } 1 \leq i \leq s\}.$$

Separating hyperplanes for polytopes

Since we assume that the polytope is given by a set of constraints described as above, the computation of separating hyperplanes is trivial. Every vector $y \in \mathbb{R}^n$ which is not contained in the polytope violates at least one constraint, i.e., there exists an index i_0,

$1 \leq i_0 \leq s$, such that $\langle a_{i_0}, y\rangle > \beta_{i_0}$. This constraint defines a hyperplane that separates y from the polytope P.

In Section 2.2.3 of Chapter 2, we considered some properties of polytopes. We can use these results obtained there to determine the corresponding parameters for the rounding method.

Properties of polytopes

In Lemma 2.2.20 in Chapter 2, we have seen that a full-dimensional polytope given by integral constraints is contained in a Euclidean ball with radius $R_{out} = n^{(n+1)/2} r^n$ centered at the origin, where r is an upper bound on the size of the polytope.

Furthermore, we have shown in Chapter 2 a lower bound for the volume of symmetric full-dimensional polyhedra, see Lemma 2.2.17. Since we want to construct an algorithm that computes an approximate Löwner-John ellipsoid also for a non-symmetric full-dimensional polytope, we need to generalize this result. To compute a lower bound for the volume of a non-symmetric full-dimensional polytope, the idea is to construct a simplex which is fully contained in the polytope. Then, the volume of this simplex provides a lower bound on the volume of the polytope.

Lemma 7.3.1. *Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope given by s integral inequalities $\langle a_i, x\rangle \leq \beta_i$, where $a_i \in \mathbb{Z}^n$, $\beta_i \in \mathbb{Z}$ for $1 \leq i \leq s$, i.e.,*

$$P = \{x \in \mathbb{R}^n | \langle a_i, x\rangle \leq \beta_i \text{ for } 1 \leq i \leq s\} = \{x \in \mathbb{R}^n | A^T x \leq b\},$$

where A is the matrix which consists of the columns a_i and $\beta := (\beta_1, \ldots, \beta_s)^T \in \mathbb{Z}^s$. Then the volume of the polytope P is at least

$$\mathrm{vol}_n(P) \geq 2^{-n^2} n^{-n(n+1)/2} \cdot r^{-n(n+1)},$$

where r is the size of the polytope.

Proof. Since P is a full-dimensional polytope, it contains $n+1$ affine independent vertices $\{v_0, \ldots, v_n\}$. The convex hull of these vertices is a simplex, which is completely contained in P, that means

$$\mathrm{vol}_n(P) \geq \mathrm{vol}_n(\mathrm{conv}(v_0, v_1, \ldots, v_n)).$$

The volume of this simplex is given by

$$\frac{1}{n!}\left|\det\begin{pmatrix} 1 & \ldots & 1 \\ v_0 & \ldots & v_n \end{pmatrix}\right|.$$

For each vertex v_i, $0 \leq i \leq n$, there exists a submatrix A_i of A^T such that $A_i v_i = d_i$, where d_i is the vector which consists of the corresponding coefficients of the vector b. Using Cramer's Rule, the j-th coefficient of v_i is of the form

$$v_{ij} = \frac{\det(A_{ij})}{\det(A_i)},$$

where A_{ij} is the matrix A_i where the j-th column is replaced by d_i. Using this, we get

$$\begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix} = \frac{1}{\prod_{i=1}^{n} \det(A_i)} \begin{pmatrix} \det(A_0) & \dots & \det(A_n) \\ \det(A_0) \cdot v_0 & \dots & \det(A_n) \cdot v_n \end{pmatrix}.$$

The matrix on the right has integral coefficients. Hence, the determinant of this matrix is at least 1,

$$\begin{aligned} \left| \det \begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix} \right| &= \frac{1}{\prod_{i=1}^{n} |\det(A_i)|} \left| \det \begin{pmatrix} \det(A_0) & \dots & \det(A_n) \\ \det(A_0) \cdot v_0 & \dots & \det(A_n) \cdot v_n \end{pmatrix} \right| \\ &\geq \frac{1}{\prod_{i=1}^{n} |\det(A_i)|}. \end{aligned}$$

Using the upper bound for the determinant from Claim 2.2.18 in Chapter 2, we get

$$|\det(A_i)| \leq n^{n/2} \operatorname{size}(A_i)^n \leq n^{n/2} r^n$$

and

$$\begin{aligned} \frac{1}{n!} \left| \det \begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix} \right| &\geq \frac{1}{n!} \left(\prod_{i=0}^{n} n^{n/2} r^n \right)^{-1} \\ &\geq \frac{1}{n!} \left(n^{n/2} r^n \right)^{-(n+1)}. \end{aligned}$$

Using $n! \leq 2^{n^2}$, see Section A.0.3 in the Appendix, we get the following lower bound for the volume of the polytope

$$\operatorname{vol}_n(P) \geq 2^{-n^2} \cdot n^{-n(n+1)/2} \cdot r^{-n(n+1)}.$$

□

7.3.2. Description and analysis of the algorithm

Obviously, it is possible to perform the algorithm that computes approximate Löwner-John ellipsoids for polytopes in the same way as the algorithm that computes approximate Löwner-John ellipsoids for ℓ_p-bodies. For polytopes, it is even possible to improve the approximation factor of the computed approximate Löwner-John ellipsoid. With these improvements, we are able to compute an approximate Löwner-John ellipsoid with approximation factor $1/\gamma$ in polynomial time instead of computing an approximate Löwner-John ellipsoid with approximation factor $2/\gamma$ in single exponential time.

Since a polytope is the intersection of finitely many halfspaces, we are able to check efficiently whether the shrinked ellipsoid $\gamma \star E$ is contained in the polytope P. Suppose the polytope P is given by s integral inequalities $\langle a_i, x \rangle \leq \beta_i$, where $a_i \in \mathbb{Z}^n$, $\beta_i \in \mathbb{Z}$ for $1 \leq i \leq s$. Then, the ellipsoid $\gamma \star E$ is contained in P if it is contained in all the halfspaces $\{x \in \mathbb{R}^n | \langle x, a_i \rangle \leq \beta_i\}$. That means, for all $1 \leq i \leq s$, it is sufficient and necessary that $\max\{\langle a_i, x\rangle | x \in \gamma \star E\} \leq \beta_i$, that means

$$\gamma \star E \subseteq P \text{ if and only if } \max\{\langle a_i, x\rangle | x \in \gamma \star E\} \leq \beta_i \text{ for all } 1 \leq i \leq s.$$

The linear function $\langle a_i, x\rangle$ has the maximum value $\langle a_i, x\rangle + \gamma\sqrt{a_i^T D a_i}$ over $\gamma \star E$, as we have seen in Lemma 6.4.4 in Chapter 6. Hence, the ellipsoid $\gamma \star E$ is contained in the polytope E if and only if $\langle a_i, x\rangle + \gamma\sqrt{a_i^T D a_i} \leq \beta_i$ for all $1 \leq i \leq s$, or equivalently if

$$\langle a_i, x\rangle \leq \beta_i - \gamma\sqrt{a_i^T D a_i} \text{ for all } 1 \leq i \leq s.$$

Geometrically, this condition can be interpreted as follows: The ellipsoid $\gamma\star E$ is contained in the polytope if and only if the vector c is contained in the shrinked polytope

$$P' = \left\{ x \in \mathbb{R}^n \,\middle|\, \langle a_i, x\rangle \leq \beta_i - \gamma\sqrt{a_i^T D a_i} \text{ for all } 1 \leq i \leq s \right\}.$$

We will prove this result formally in the following lemma.

Lemma 7.3.2. *Let $P \subset \mathbb{R}^n$ be a full-dimensional polytope given by s integral inequalities $\langle a_i, x\rangle < \beta_i$, where $a_i \in \mathbb{Z}^n$, $\beta_i \in \mathbb{Z}$ for $1 \leq i \leq s$, i.e., $P = \{x \in \mathbb{R}^n | \langle a_i, x\rangle \leq \beta_i$ for all $1 \leq i \leq s\}$. Let $0 < \gamma < 1/n$. Let $E = E(D, c)$ be an ellipsoid in $\mathbb{R}^n$. If the center of this ellipsoid is contained in the shrinked polytope*

$$P' := \{x \in \mathbb{R}^n | \langle a_i, x\rangle \leq \beta_i - \gamma\sqrt{a_i^T D a_i} \textit{ for all } 1 \leq i \leq s\} \tag{7.15}$$

the ellipsoid $\gamma \star E$ is contained in the polytope, $\gamma \star E \subseteq P$.

To prove the lemma, we can obviously argument as above, but we can also prove it directly using the generalized Cauchy-Schwarz inequality.

Proof. We consider a vector $x \in \gamma \star E = E(\gamma^2 D, c)$, i.e.,

$$(x - c)^T(\gamma^2 D)^{-1}(x - c) \leq 1 \text{ or } (x - c)^T D^{-1}(x - c) \leq \gamma^2. \tag{7.16}$$

We will show that such a vector x satisfies all s constraints defining the polytope P. Let $1 \leq i \leq s$. Using the generalized Cauchy-Schwarz-inequality for symmetric positive definite matrices, see Lemma 2.2.5 in Chapter 2, we obtain

$$\langle x, a_i\rangle = \langle x - c, a_i\rangle + \langle c, a_i\rangle \leq \sqrt{(x - c)^T D^{-1}(x - c)} \cdot \sqrt{a_i^T D a_i} + \langle c, a_i\rangle.$$

Since x is an element from the ellipsoid $\gamma \star E$, (7.16), and since the vector c is contained in the polytope P' defined in (7.15), this is at most

$$\langle x, a_i\rangle \leq \sqrt{\gamma^2} \cdot \sqrt{a_i^T D a_i} + \beta_i - \gamma\sqrt{a_i^T D a_i} = \beta_i.$$

□

Now, we are able to present a detailed description of the algorithm that computes an approximate Löwner-John ellipsoid for full-dimensional polytopes. As in the general rounding method for convex sets, the algorithm computes in the initialization step a radius R_{out} such that $P \subseteq \bar{B}_n^{(2)}(0, R_{out})$ according to Lemma 2.2.20 in Chapter 2. This ball is chosen as the initial ellipsoid. After this, the algorithm works iteratively. Given an ellipsoid $E(D_k, c_k)$, the algorithm considers a shrinked polytope

$$P' = \{x \in \mathbb{R}^n | \langle x, a_i \rangle \leq \beta_i - \gamma \sqrt{a_i^T D_k a_i} \text{ for all } 1 \leq i \leq s\}$$

and it checks if the center c_k of the ellipsoid is contained in this polytope. If this is the case the ellipsoid $\gamma \star E(D_k, c_k)$ is contained in the polytope as we have seen in Lemma 7.3.2.

Otherwise, there exists an index i_0 such that the condition is violated, i.e., $\langle c_k, a_{i_0} \rangle > \beta_{i_0} - \gamma \sqrt{a_{i_0}^T D_k, a_{i_0}}$. For such an index, the algorithm considers the intersection of the ellipsoid E_k and the halfspace $\{x \in \mathbb{R}^n | \langle a_{i_0}, x \rangle \leq \langle c_k, a_{i_0} \rangle + \gamma \sqrt{a_{i_0}^T D_k a_{i_0}}\}$ and constructs an ellipsoid E_{k+1} which contains this intersection according to the construction in Section 7.1.2. A detailed description of the algorithm is given in Algorithm 21.

The rounding method for polytopes is a variant of the rounding method for bounded convex sets, but we are not able to transfer the results directly since we use another criterion to decide whether we have already found an approximate Löwner-John ellipsoid. Especially, we need to show that we can construct a shallow cut if we have not found an approximate Löwner-John ellipsoid.

In the next lemma, we state the main properties of the algorithm. We show that each ellipsoid constructed by the algorithm satisfies the property that it contains the polytope. Additionally, we show that the output of the algorithm is an approximate Löwner-John ellipsoid.

Lemma 7.3.3. *Let $0 < \gamma \leq 1/n$. Let $P \subseteq \mathbb{R}^n$ be a full-dimensional polytope given by s integral inequalities $\langle a_i, x \rangle \leq \beta_i$, where $a_i \in \mathbb{Z}^n$, $\beta_i \in \mathbb{Z}$ for $1 \leq i \leq s$, i.e., $P = \{x \in \mathbb{R}^n | \langle a_i, x \rangle \leq \beta_i$ for all $1 \leq i \leq s\}$. Then, the rounding algorithm for polytopes, Algorithm 21, satisfies the following properties:*

- *Each ellipsoid E_k constructed by the algorithm contains the polytope, $P \subseteq E(D_k, c_k)$ for all $k \geq 0$.*
- *The output of the algorithm is a $1/\gamma$-approximate Löwner-John ellipsoid E of P, that means*

$$E \subseteq P \subseteq (1/\gamma) \star E.$$

Proof. First, we show that every ellipsoid which is constructed by the algorithm contains the polytope P. We do this by induction in the same way as in the proof of Lemma 7.1.13 in Section 7.1.

Algorithm 21 Rounding method for polytopes

Input:

- A full-dimensional polytope defined by s constraints $\langle a_i, x\rangle \leq \beta_i$, where $a_i \in \mathbb{Z}^n, \beta_i \in \mathbb{Z}$, $1 \leq i \leq s$, and
- a parameter γ with $0 < \gamma < 1/n$.

Output: An ellipsoid $E \subseteq \mathbb{R}^n$ given by a symmetric positive definite matrix D and a center c.

1. Set $r \leftarrow \max\{\text{size}(a_i), \text{size}(\beta_i) | 1 \leq i \leq s\}$.
2. (Initialization)
   Set
   a) $R_{out} \leftarrow n^{(n+1)/2} r^n$,
   b) $N \leftarrow \lceil 2(n+1)^3 n \left(\log_2(R_{out}) + (n+1)\log_2(2nr)\right)\rceil$ and
   c) $D_0 \leftarrow R^2 \cdot I_n$ and $c_0 \leftarrow 0$.
3. For $0 \leq k \leq N$, check if there exists an index i_0, $1 \leq i_0 \leq s$ such that
   $$\langle c_k, a_{i_0}\rangle > \beta_{i_0} - \gamma\sqrt{a_{i_0}^T D_k a_{i_0}}.$$
   a) If no such inequality exists, output $E(\gamma^2 D_k, c_k)$.
   b) Otherwise, set
   $$c_{k+1} \leftarrow c_k - \frac{1-n\gamma}{n+1}\frac{D_k a_{i_0}}{\sqrt{a_{i_0}^T D_k a_{i_0}}} \quad \text{and}$$
   $$D_{k+1} \leftarrow \frac{n^2(1-\gamma^2)}{n^2-1}\left(D_k - \frac{2(1-n\gamma)}{(n+1)(1-\gamma)}\frac{D_k a_{i_0}(D_k a_{i_0})^T}{a_{i_0}^T D_k a_{i_0}}\right).$$

In the initialization, the algorithm computes the parameter R_{out} as a radius of a circumscribed Euclidean ball, $P \subseteq \bar{B}_n^{(2)}(0, R_{out}) = E(R_{out}^2 I_n, 0)$, see Lemma 2.2.20.
If we consider an index $k > 0$ such that $P \subseteq E(D_k, c_k)$, the algorithm constructs the ellipsoid $E_{k+1} = E(D_{k+1}, c_{k+1})$ only if there exists an index i_0, $1 \leq i_0 \leq m$, such that

$$\langle c_k, a_{i_0}\rangle > \beta_{i_0} - \gamma\sqrt{a_{i_0}^T D_k a_{i_0}}.$$

In this case, every element $x \in P$ satisfies

$$\langle a_{i_0}, x\rangle \leq \beta_{i_0} < \langle c_k, a_{i_0}\rangle + \gamma\sqrt{a_{i_0}^T D_k a_{i_0}}.$$

That means P is contained in the halfspace

$$\left\{x \in \mathbb{R}^n | \langle a_{i_0}, x\rangle \leq \langle c_k, a_{i_0}\rangle + \gamma\sqrt{a_{i_0}^T D_k a_{i_0}}\right\}.$$

Since we assume that P is also contained in the ellipsoid $E(D_k, c_k)$, we have

$$P \subseteq E(D_k, c_k) \cap \left\{x \in \mathbb{R}^n | \langle a_{i_0}, x\rangle \leq \langle c_k, a_{i_0}\rangle + \gamma\sqrt{a_{i_0}^T D_k a_{i_0}}\right\}.$$

According to Theorem 7.1.8 with the parameter $\zeta = \gamma$, the ellipsoid $E(D_{k+1}, c_{k+1})$ is defined such that it contains the intersection of the ellipsoid E_k with the halfspace,

$$E_k \cap \{x \in \mathbb{R}^n | \langle a_{i_0}, x\rangle \leq \langle c_k, a_{i_0}\rangle + \gamma\sqrt{a_{i_0}^T D_k a_{i_0}}\} \subseteq E_{k+1},$$

which shows that $P \subseteq E_{k+1}$. The algorithm terminates after k iterations if the center c_k of the ellipsoid E_k is contained in the shrinked polytope P', that means if

$$\langle c_k, a_i\rangle \leq \beta_i - \gamma\sqrt{a_i^T D_k a_i} \text{ for all } 1 \leq i \leq s. \tag{7.17}$$

As we have seen in Lemma 7.3.2, this guarantees that $\gamma \star E_k \subseteq P$. Altogether, the ellipsoid E_k satisfies

$$\gamma \star E_k \subseteq P \subseteq E_k.$$

The algorithm outputs the symmetric positive definite matrix $D = \gamma^2 D_k$ and the vector $c = c_k$. The ellipsoid $E(D, c)$ defined by this matrix D and this vector c satisfies

$$E(D, c) = E(\gamma^2 D_k, c_k) \subseteq P \subseteq E(D_k, c_k) = \frac{1}{\gamma} \star E(\gamma^2 D_k, c_k) = \frac{1}{\gamma} \star E(D, c).$$

Hence, the ellipsoid $E(D, c)$ is a $1/\gamma$-approximate Löwner-John ellipsoid. □

It remains to show that the algorithm really outputs an ellipsoid. This can be done analogously to the general rounding method for bounded convex sets using that the volume of each constructed ellipsoid decreases by a single exponential factor. In Lemma 7.3.1, we have seen that the volume of the polytope P is at least

$$\begin{aligned} \mathrm{vol}_n(P) &\geq 2^{-n^2} n^{-n(n+1)/2} r^{-n(n+1)} \qquad (7.18)\\ &\geq 2^{-n^2} n^{-n(n+1)/2} r^{-n(n+1)} \frac{\mathrm{vol}_n(B_n^{(2)}(0,1))}{\mathrm{vol}_n(B_n^{(\infty)}(0,1))}\\ &= 2^{-n(n+1)} n^{-n(n+1)/2} r^{-n(n+1)} \mathrm{vol}_n(B_n^{(2)}(0,1))\\ &\geq (2nr)^{-n(n+1)} \mathrm{vol}_n(B_n^{(2)}(0,1)). \end{aligned}$$

Hence, $r_{in} = (2nr)^{-(n+1)}$ provides a lower bound on the volume of the polytope P. Consequently, in the initialization step of the algorithm, the upper bound for the number of iterations is chosen as

$$N = 2(n+1)^3 n \left(\log_2(R_{out}) + (n+1)\log_2(2nr)\right),$$

where R_{out} is the radius of the circumscribed Euclidean ball, which defines the initial ellipsoid.
Combining these results, we obtain the following theorem.

Theorem 7.3.4. *(Theorem 6.4.12 restated.)*
Given a full-dimensional, bounded polytope $P = \{x \in \mathbb{R}^n | \langle a_i, x \rangle \leq \beta_i$ for all $1 \leq i \leq s\}$ with $a_i \in \mathbb{Z}^n, \beta_i \in \mathbb{Z}$, and a parameter γ with $0 < \gamma < 1/n$, the rounding method for polytopes, Algorithm 21, computes a $1/\gamma$-approximate Löwner-John ellipsoid, i.e., a positive definite matrix $D \in \mathbb{Q}^{n \times n}$ and a vector $c \in \mathbb{Q}^n$ defining the ellipsoid $E(D, c)$ such that

$$E(D, c) \subseteq P \subseteq \frac{1}{\gamma} \star E(D, c).$$

The algorithm runs in polynomial space and the number of arithmetic operations of the algorithm is

$$(ns \cdot \log_2(r))^{\mathcal{O}(1)}$$

where r is the size of the polytope. The size of the approximate Löwner-John ellipsoid is at most

$$2^{\mathcal{O}(n^4)} r^{\mathcal{O}(n)}.$$

Proof. As we have seen in Lemma 7.3.3, if the algorithm outputs an ellipsoid $E(D, c)$ then this ellipsoid satisfies

$$E(D, c) \subseteq P \subset \frac{1}{\gamma} \star E(D, c).$$

So far, we have not shown that it is guaranteed that the algorithm outputs something. Hence, we assume that the algorithm constructs all N ellipsoids E_N. In this case the algorithm would not output anything. The number of iterations of the algorithm is N, where

$$N \geq 2n(n + 1)^3 (\log_2(R_{out}) + (n + 1) \log_2(2nr)).$$

We have seen in (7.18) that

$$\operatorname{vol}_n(P) \geq (2nr)^{-n(n+1)} \operatorname{vol}_n(B_n^{(2)}(0, 1)).$$

Hence, it follows from Lemma 7.1.14 that after N iterations the volume of the ellipsoid constructed in the N-th iteration is less than the volume of the polytope,

$$\operatorname{vol}_n(E_N) < \operatorname{vol}_n(P).$$

This is a contradiction to the fact that each ellipsoid constructed by the algorithm contains the polytope as we have proven in Lemma 7.3.3.
The number of arithmetic operations of the algorithm is dominated by the number of

iterations, which is at most N. In each iteration, we need to check s constraints. This can be done using at most $n^{\mathcal{O}(1)}$ arithmetic operations. Also the rest of the computations can be done using at most $(n \cdot s)^{\mathcal{O}(1)}$ arithmetic operations. Hence, we obtain the following upper bound for the number of arithmetic operations of the rounding method for polytopes

$$\begin{aligned} & \left(2n(n+1)^3(\log_2(R_{out}) + (n+1)\log_2(2nr))\right)(sn)^{\mathcal{O}(1)} \\ = & \left(2n(n+1)^3(\log_2(\sqrt{n}n^{n/2}r^n) + (n+1)\log_2(2nr))\right) \cdot (sn)^{\mathcal{O}(1)} \\ \leq & \ (ns\log_2(r))^{\mathcal{O}(1)}. \end{aligned}$$

As we have seen in Theorem 7.1.18, we can assume that the algorithm runs in polynomial space and that the size of each constructed instance $E_k = E(D_k, c_k)$ is at most

$$2^{\mathcal{O}(n^4)}(R_{out}r_{in}^{-1})^{\mathcal{O}(1)}.$$

Since $R_{out} = n^{(n+1)/2}r^n$ and $r_{in}^{-1} = (2nr)^{n+1}$, this is upper bounded by

$$2^{\mathcal{O}(n^4)}(n^{(n+1)/2}r^n(2nr)^{n+1})^{\mathcal{O}(1)} = 2^{\mathcal{O}(n^4)}r^{\mathcal{O}(n)}.$$

□

7.4. Discussion of the results

In this chapter, we have described algorithms that compute approximate Löwner-John ellipsoids for the class of ℓ_p-bodies with $1 < p < \infty$ and for polytopes. Hence, our assumptions made in Chapter 6 are satisfied and there exists a deterministic polynomially space bounded algorithm that solves the lattice membership problem for ℓ_p-balls and polytopes. As we have seen in Theorem 4.3.13 in Chapter 4 this leads to a deterministic polynomially space bounded algorithm that solves the closest vector problem with respect to an ℓ_p-norm, $1 < p < \infty$, or a polyhedral norm, e.g. the ℓ_1-norm or the ℓ_∞-norm.

We presented the algorithms by using a general framework which computes for a bounded convex set given by a separation oracle a $2/\gamma$-approximate Löwner-John ellipsoid for some parameter $0 < \gamma < 1/n$. The number of arithmetic operations and the number of calls to the oracle are polynomial in $1/\gamma$, but single exponential in the dimension n. This general framework could be adapted to the class of ℓ_p-bodies such that we obtain an algorithm which computes a $2/\gamma$-approximate Löwner-John ellipsoid for a given ℓ_p-body, where the number of arithmetic operations of the algorithm is polynomial in $1/\gamma$, logarithmic in the size of the ℓ_p-body, and single exponential in the dimension.

With regard to the approximation factor, this result is almost optimal, since for every full-dimensional bounded convex set, there exists a n-approximate Löwner-John ellipsoid as it is proven in John's lemma. On the other hand, an improvement of the running time would be desirable but seems to be impossible using the techniques presented in Section

7.1. For our applications, the single exponential running time is negligible, since the running time of the lattice membership algorithm is mainly influenced by the approximation factor of the computed Löwner-John ellipsoid.

That it is possible to improve our result for concrete classes of bounded convex sets show the results of Goffin and Lenstra, which we presented in Section 7.3. They showed that for the class of polytopes, there exists a polynomial time algorithm that computes for a given polytope in $\mathbb{R}^n$ an approximate Löwner-John ellipsoid with approximation factor $\mathcal{O}(n)$.

A. Appendix

A.0.1. Hadamard's inequality

Let $B = [b_1, \ldots, b_m] \in \mathbb{R}^{n \times m}$ with $b_1, \ldots, b_m$ linearly independent. Then we have

$$\sqrt{B^T \cdot B} \leq \prod_{i=1}^{m} \|b_i\|_2$$

where equality holds if and only if the vectors $b_1, \ldots, b_m$ are orthogonal. In particular, if $B \in \mathbb{R}^{n \times n}$, then

$$|\det(B)| \leq \prod_{i=1}^{n} \|b_i\|_2.$$

A.0.2. Chebyshev's inequality

Theorem A.0.1. *(Chebyshev's inequality)*
Let X be a random variable of finite expectation and $\delta > 0$ fixed. Then

$$\Pr[|X - E(X)| \geq \delta] \leq \frac{\mathrm{Var}(X)}{\delta^2}.$$

For a proof of this inequality see for example [CA06].

A.0.3. The Gamma function and Stirling's formula

For $x \in \mathbb{R}$, $x > 0$, the Gamma Function is defined as

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

For $n \in \mathbb{N}$, we have $\gamma(n) = (n-1)!$ and one can show that for all $x \in \mathbb{R}$ we have $\Gamma(x+1) = x \cdot \Gamma(x)$.
Using Stirling's formula, we obtain that

$$\Gamma(x) = \sqrt{\frac{2\pi}{x}} \left(\frac{x}{e}\right)^x e^{\nu(x)},$$

where ν is a function satisfying $0 < \nu(x) < 1/(12x)$ for all $x \in \mathbb{R}$.

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n < n! < e^{1/(12n)}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

Bibliography

[ABSS93] Sanjeev Arora, László Babai, Jacques Stern, and Elizabeth Sweedyk. The hardness of approximate optimia in lattices, codes, and systems of linear equations. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 724 – 733, 1993.

[AJ08] Vikraman Arvind and Pushkar S. Joglekar. Some sieving algorithms for lattice problems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 25 – 36. Schloss Dagstuhl - Leibnitz-Zentrum für Informatik, 2008.

[Ajt98] Miklós Ajtai. The shortest vector problem in ℓ_2 is NP-hard for randomized reductions. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, pages 10 – 19. Association for Computing Machinery, 1998.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33th ACM Symposium on Theory of Computing (STOC)*, pages 601 – 610. Association for Computing Machinery, 2001.

[AKS02] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Proceedings of the 17th IEEE Annual Conference on Computational Complexity (CCC)*, pages 53 – 57, 2002.

[Bab86] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1 – 13, 1986.

[Bal97] Keith Ball. An elementary introduction to modern convex geometry. In Silvio Levy, editor, *Flavors of Geometry*, volume 31 of *MSRI Publications*, pages 1 – 58. Springer Verlag, 1997.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625 – 635, 1993.

[Bar02] Alexander Barvinok. *A Course in Convexity.* American Mathematical Society, 2002.

[BJWW98] Alexander Barvinok, Davis Johnson, Gerhard Woeginger, and Russell Woodroofe. The maximum traveling salesman problem under polyhedral norms. In *Proceedings of the 6th International Integer Programming and Combinatorial Optimization Conference (IPCO)*, volume 1412 of *Lecture Notes in Computer Science*, pages 195 – 201, 1998.

[Blö00] Johannes Blömer. Closest vectors, successive minima, and dual HKZ-bases of lattices. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1853 of *Lecture Notes in Computer Science*, pages 248 – 259. Springer Verlag, 2000.

[BLPS99] Wojciech Banaszczyk, Alexander E. Litvak, Alain Pajor, and Stanislaw J. Szarek. The flatness theorem for nonsymmetric convex bodies via the local theory of banach spaces. *Mathematics of Operations Research*, 24(3):728 – 750, 1999.

[BN07] Johannes Blömer and Stefanie Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. In *Proceedings of the 34th International Colloquium of Automata, Languages and Programming (ICALP)*, volume 4596 of *Lecture Notes in Computer Science*, pages 65 – 77. Springer Verlag, 2007.

[BN09] Johannes Blömer and Stefanie Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. *Theoretical Computer Science*, 410(18):1648 – 1665, 2009. Special issue on the 34th International Colloquium of Automata, Languages and Programming (ICALP) 2007.

[BN11] Johannes Blömer and Stefanie Naewe. Solving the closest vector problem with respect to ℓ_p norms. *Computing Research Repository (CoRR)*, 2011. `arxiv:1104.3720 [cs.DS]` Submitted to publication.

[BS99] Johannes Blömer and Jean-Pierre Seifert. The complexity of computing short linearly independent vectors and short bases in a lattice. In *Proceedings of the 31th Symposium on Theory of Computing (STOC)*, pages 711 – 720. Association for Computing Machinery, 1999.

[BV09] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2009.

[CA06] Kai L. Chung and Farid AitSahlia. *Elementary Probability Theory: With stochastic processes and an introduction to mathematical finance.* Springer, 4 edition, 2006.

[Cas71] John W. S. Cassels. *An Introduction to the Geometry of Numbers.* Springer, 1971.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory.* Springer, 1993.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third ACM Symposium on Theory of Computing (STOC)*, pages 151 – 158. Association for Computing Machinery, 1971.

[CS93] John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups.* Springer, 3rd edition, 1993.

[DFK91] Martin Dyer, Alan M. Frieze, and Ravi Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *Journal of the ACM*, 38(1):1 – 17, 1991.

[DGK63] Ludwig Danzer, Branko Grünbaum, and Victor Klee. Helly's theorem and its relatives. *Proceedings of Symposia in Pure Mathematics*, 7:101 – 180, 1963.

[Din02] Irit Dinur. Approximating $\textsc{Svp}_\infty$ to within almost-polynomial factors is NP-hard. *Theoretical Computer Science*, 285(1):55 – 71, 2002. Special issue on the 4th Italian Conference on Algorithms and Complexity (CIAC 2000).

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205 – 243, 2003.

[DKS98] Irit Dinur, Guy Kindler, and Shmuel Safra. Approximating $\textsc{Cvp}$ to within almost-polynomial factors is NP-hard. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 99 – 111. IEEE Computer Society, 1998.

[DPV10] Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative algorithms for the shortest and closest lattice vector problems in any norm via M-ellipsoid coverings. *Computing Research Repository (CoRR)*, 2010. `arxiv:1011.5666 [cs.DS]`.

[DPV11] Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative lattice algorithms in any norm via M-ellipsoid coverings. In *Proceedings of the 52th Annual Symposium on Foundations of Computer Science (FOCS).* IEEE Computer Society, 2011. To Appear. For a preprint see [DPV10].

[DV11] Daniel Dadush and Santosh Vempala. Deterministic construction of an approximate M-ellipsoid and its application to derandomizing lattice algorithms. *Computing Research Repository (CoRR)*, 2011. `arXiv:1107.5478v1 [cs.CC]`.

[DV12] Daniel Dadush and Santosh Vempala. Deterministic construction of an approximate M-ellipsoid and its application to derandomizing lattice algorithms. In *Proceedings of the 23th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).* Society for Industrial and Applied Mathematics, 2012. To Appear. For a preprint see [DV11].

[EHN11] Friedrich Eisenbrand, Nicolai Hähnle, and Martin Niemeier. Covering cubes and the closest vector problem. In *Proceedings of the 27th Symposium on Computational Geometry (SoCG)*, pages 417 – 423. American Mathematical Society, 2011.

[Eis10] Fritz Eisenbrand. Integer programming and algorithmic geometry of numbers. In Michael Jünger, Thomas Liebling, Denis Naddef, George Nemhauser, William Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence Wolsey, editors, *50 Years of Integer Programming 1958 - 2008*. Springer, 2010.

[Fru76a] Michael Frumkin. Algorithms for the solution in integers of systems of linear equations. *Issledovaniya po diskretnoi optimizatsii (Studies in Discrete Optimization)*, pages 96 – 127, 1976. In Russian.

[Fru76b] Michael Frumkin. An application of modular arithmetic to the construction of algorithms for solving systems of linear equations. *Doklady Akademii Nauk SSSR*, 229:1067 – 1070, 1976. In Russian.

[FT87] András Frank and Éva Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7:49 – 65, 1987.

[Gau01] Carl Friedrich Gauß. *Disquisitiones Arithmeticae.* Gerhard Fleischer, Leipzig, 1801.

[GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60:540 – 563, 2000.

[GL81] Peter Gács and László Lovász. Khachiyan's algorithm for linear programming. *Mathematical Programming Study*, 14:61 – 68, 1981.

[GLS93] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometrix Algorithms and Combinatorial Optimization.* Springer, 2nd edition, 1993.

[GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55 – 61, 1999.

[Gof84] Jean-Louis Goffin. Variable metric relaxation methods, part II: The ellipsoid method. *Mathematical Programming*, 30:147 – 162, 1984.

[Hås88] Johan Håstad. Dual vectors and lower bounds for the nearest lattice point problem. *Combinatorica*, 8(1):75 – 81, 1988.

[Hei05] Sebastian Heinz. Complexity of integer quasiconvex polynomial optimization. *Journal of Complexity*, 21(4):543 – 556, 2005.

[Hel85] Bettina Helfrich. Algorithms to construct Minkowski reduced and Hermite reduced lattice bases. *Theoretical Computer Science*, 41:125 – 139, 1985.

[Hil11] David Hilbert, editor. *Gesammelte Abhandlungen von Hermann Minkowski.* Chelsea Publishing Company, 1911.

[HJ85] Roger A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, 1985.

[HK10] Robert Hildebrand and Matthias Köppe. A faster algorithm for quasi-convex integer polynomial optimization. *Computing Research Repository (CoRR)*, 2010. `arXiv:1006.4661 [math.OC]`.

[HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC)*, pages 469 – 477. Association for Computing Machinery, 2007.

[HS07] Guillaume Hanrot and Damien Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In *Proceedings of the 27th Annual International Cryptology Conference (Crypto)*, volume 4622 of *Lecture Notes in Computer Science*, pages 170 – 186. Springer, 2007.

[Joh48] Fritz John. Extremum problems with inequaliteis as subsidiary conditions. In *Studies and Essays, presented to R. Courant on his 60th Birthday, January 8, 1948*, pages 187 – 204. Interscience, 1948.

[Kan87a] Ravi Kannan. Algorithmic geometry of numbers. *Annual Reviews in Computer Science*, 2:231 – 267, 1987.

[Kan87b] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415 – 440, 1987.

[Khi48] A. Ya. Khinchin. A quantitative formulation of kronecker's theory of approximation. *Izvestiya Akademii Nauk SSR Seriya Matematika*, (12):113 – 122, 1948. In Russian.

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789 – 808, 2005.

[Kho10] Subhash Khot. Inapproximability results for computational problems on lattices. In Phong Q. Nguyen and Brigitte Vallée, editors, *The LLL Algorithm - Survey and Applications.* Springer, 2010.

[Kle00] Philip Klein. Finding the closest vector when it is unusually close. In *Proceedings of 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 937 – 941. Society for Industrial and Applied Mathematics, 2000.

[KLS97] Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an $\mathcal{O}^*(\mathrm{n}^5)$ volume algorithm for convex bodies. *Random Struct. Algorithms*, 11(1):1 – 50, 1997.

[Koc94] Martin Kochol. Constructive approximation of a ball by polytopes. *Mathematic Slovaca*, 44(1):99 – 105, 1994.

[KS96] Michael Kaib and Claus-Peter Schnorr. The generalized gauss reduction algorithm. *Journal of Algorithms*, 21(3):565 – 578, 1996.

[Len83] Hendrik W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538 – 548, 1983.

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515 – 534, 1982.

[LLS90] Jeffrey C. Lagarias, Hendrik W. Lenstra, and Claus-Peter Schnorr. Korkinzolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333 – 348, 1990.

[Lov86] László Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity.* Society For Industrial And Applied Mathematics, 1986.

[LS92] László Lovász and Herbert E. Scarf. The generalized basis reduction algorithm. *Mathematics of Operations Research*, 17(3):751 – 764, 1992.

[Man99] Olvi Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24(1–2):15–23, 1999.

[Mar03] Jacques Martinet. *Perfect lattices in Euclidean Spaces.* Springer, 2003.

[Mat02] Jirí Matousek. *Lectures on Discrecte Geometry.* Springer, 2002.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems - A Cryptographic Perspective.* Kluwer Academic Publishers, 2002.

[Mic01] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, 2001.

[Mic07] Daniele Micciancio. Lecture note on lattice algorithms and applications, lecture 7: Svp, Cvp and minimum distance, 2007.

[Mic08] Daniele Micciancio. Efficient reductions among lattice problems. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 84 – 93. Society for Industrial and Applied Mathematics, 2008.

[MN99] Jan R. Magnus and Heinz Neudecker. *Matrix Differentital Calculus with Applications in Statistics and Econometrics.* John Wiley & Sons, 2nd edition, 1999.

[MV10a] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In *Proceedings of the 42th ACM Symposium on Theory of Computing (STOC)*, pages 351 – 358. Association for Computing Machinery, 2010.

[MV10b] Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *Proceedings of the 21th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1468 – 1480. Society for Industrial and Applied Mathematics, 2010.

[Ngu01] Phong Q. Nguyen. The dark side of the hidden number problem: Lattice attacks on DSA. In *Cryptography and Computational Number Theory*, volume 20 of *Progress in Computer Science and Applied Logic.* Birkhäuser, 2001. Proceedings of the CCNT Workshop in Singapore, 1999.

[Nie07] Martin Niemeier. Reduktionen von Cvp mit wenigen Lösungen auf Cvp mit eindeutigen Lösungen, 2007. Bachelor thesis.

[NS00] Phong Q. Nguyen and J. Stern. Lattice reduction in cryptology - an update. In *Proceedings of the 4th Algorithmic Number Theory Symposium (ANTS IV)*, number 1838 in Lecture Notes in Computer Science, pages 85 – 112. Springer, 2000.

[NV08] Phong Q. Nguyen and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2), 2008.

[NV10] Phong Q. Nguyen and Brigitte Vallée, editors. *The LLL-Algorithm - Survey and Applications.* Springer, 2010.

[Pei08] Chris Peikert. Limits on the hardness of lattice problems in ℓ_p norms. *Computational Complexity*, 17(2):300 – 351, 2008. Special issue on CCC 2007.

[Pol87] Boris T. Polyak. *Introduction to Optimization.* Optimization Software, 1987.

[PS98] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization.* Dover Publications, 1998.

[PS09] Xavier Pujol and Damien Stehle. Solving the shortest lattice vector problem in time $2^{2.465}n$. Cryptology ePrint Archive, Report 2009/605, 2009.

[Reg04] Oded Regev. Lecture note on lattices in computer science, lecture 8: $2^{O(n)}$-time algorithm for SVP, 2004.

[Reg10] Oded Regev. On the complexity of lattice problems with polynomial approximation factors. In Phong Q. Nguyen and Brigitte Vallée, editors, *The LLL Algorithm - Survey and Applications*. Springer, 2010.

[Rit96] Harald Ritter. Breaking knapsack cryptosystems by ℓ_∞ enumeration. In *1st International Conference of the Theory and Appications of Cryptology (Pragocrypt '96)*, pages 480 – 492. CTU Publishing House, 1996.

[Roc70] Ralph T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[RR06] Oded Regev and Ricky Rosen. Lattice problems and norm embeddings. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC)*, pages 447 – 456. Association for Computing Machinery, 2006.

[Rud00] Mark Rudelson. Distances between non-symmetric convex bodies and the MM^*-estimate. *Positivity*, 4(2):161 – 178, 2000.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201 – 224, 1987.

[Sch91] Wolfgang M. Schmidt. *Diophantine Approximations and Diophantine Equations*. Lecture Notes in Mathematics. Springer-Verlag, 1991.

[Sch94] Claus-Peter Schnorr. Block reduced lattice bases and successive minima. *Combinatorics, Probability & Computing*, 3:507 – 522, 1994.

[Sho77] Naum Z. Shor. Cut-off method with space extension in convex programming problems. *Kibernetika*, 1, 1977. in Russian. English translation: Cybernetics 15 (1979) 502 – 508.

[SK09] Hans Rudolf Schwarz and Norbert Köckler. *Numerische Mathematik*. Springer, 2009.

[Sme10] Ionica Smeets. The history of the LLL-algorithm. In Phong Q. Nguyen and Brigitte Vallée, editors, *The LLL-Algorithm - Survey and Applications*. Springer, 2010.

[Ste04] J. Michael Steele. *The Cauchy-Schwarz Master Class*. Cambridge University Press, 2004.

[Str06] Gilbert Strang. *Linear Algebra and its Applications*. Thomson, 4th edition, 2006.

[Vaa79] Jeffrey D. Vaaler. A geometric inequality with applications to linear forms. *Pacific Journal of Mathematics*, 83(2):543 – 553, 1979.

[Vaz01] Vijay V. Vazirani. *Approximation Algorithms.* Springer, 2001.

[vEB81] Peter van Emde Boas. Another NP - complete partition problem and the complexity of computing short vectors in a lattice. Technical Report 81 – 04, Department of Mathematics, University of Amsterdam, 1981.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, 2nd edition, 2003.

[vzGS76] Joachim von zur Gathen and Malte Sieveking. Weitere zum erfüllungsproblem polynomial äquivalente kombinatorische aufgaben. In *Komplexität von Entscheidungsproblemen*, volume 43 of *Lecture Notes in Computer Science*, pages 49 – 71. Springer, 1976.

[Web94] Roger Webster. *Convexity.* Oxford University Press, 1994.

[Ye92] Yinyu Ye. On affine scaling algorithms for nonconvex quadratic programming. *Mathematical Programming*, 56:285 – 300, 1992.

[YN76a] David B. Yudin and Arkadi S. Nemirovski. Evaluation of the information complexity of (math)ematical programming problems. *Èkonomika i Matematicheskie Metody*, 12:128 – 142, 1976. In Russian. English translation. Matekon 12 (1976), no. 2, 3 – 25.

[YN76b] David B. Yudin and Arkadi S. Nemirovskii. Informational complexity and efficient methods for the solution of convex extremal problems. *Èkonomika i Matematicheskie Metody*, 12:357 – 369, 1976. In Russian. English translation: Matekon 13 (1977), no. 3, 25 – 45.