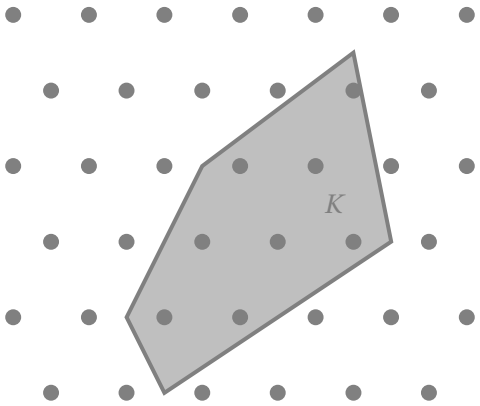
Integer Optimization and Lattices

Spring 2016

Thomas Rothvoss



UNIVERSITY of WASHINGTON

Last changes: May 25, 2016

Contents

Chapter 1

Lattices

In this chapter, we introduce the concept of lattices. Lattices are fundamentally important in discrete geometry, cryptography, discrete optimization and computer science as a whole. We follow to some extend the short, but very readable material in Chapter 2 of the text book "*Lectures on Discrete Geometry*" by Jiri Matousek [Mat02] and to some extend the excellent lecture notes by Oded Regev[1] as well as the ones by Chris Peikert[2].

1.1 Introduction to Lattices

Lattices are integral combinations of linearly independent vectors. Formally, a lattice is a set

$$\left\{ \sum_{i=1}^{k} \lambda_i \boldsymbol{b}_i \mid \lambda_1, \ldots, \lambda_k \in \mathbb{Z} \right\}$$

where $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k \in \mathbb{R}^n$ are linearly independent vectors. An alternative definition is to say that a lattice is a *discrete subgroup* of $\mathbb{R}^n$, where "discrete" means that there is an $\varepsilon > 0$ so that all points in the lattice have at least distance ε from each other.

If $k = n$, then the lattice has *full rank*. As every lattice is just a full rank lattice when restricted to the subspace $\operatorname{span}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$, most of the time we will consider full-rank lattices. In fact, we will drop the term "full-rank" and assume it implicitly from now on if not announced otherwise.

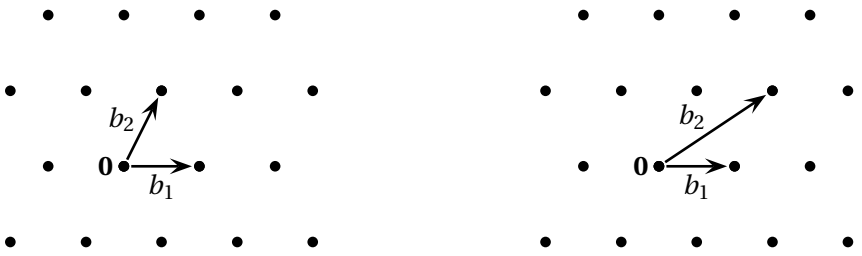
We can shortcut notation and let $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ be the matrix that has the basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ as columns. Then we abbreviate the lattice as

$$\Lambda(\boldsymbol{B}) = \left\{ \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i \mid \lambda_1, \ldots, \lambda_n \in \mathbb{Z} \right\}$$

The matrix $\boldsymbol{B}$ itself is also called a *basis* of the lattice $\Lambda(\boldsymbol{B})$. Note that a lattice has more than one basis. For example adding any integral multiple of $\boldsymbol{b}_i$ to $\boldsymbol{b}_j$ for $j \neq i$ will preserve the set of points that can be generated. Below we see an example of two different bases for the same underlying lattice:

[1] See `http://www.cims.nyu.edu/~regev/teaching/lattices_fall_2009`

[2] see `http://www.cc.gatech.edu/~cpeikert/lic13`



1.1.1 Unimodular matrices

We want to spend a bit of time characterizing the different bases for a lattice.

Definition 1. An $n \times n$ matrix $\boldsymbol{U}$ is called *unimodular*, if $\boldsymbol{U} \in \mathbb{Z}^{n\times n}$ and $\det(\boldsymbol{U}) \in \{\pm 1\}$.

Lemma 1.1. *If $\boldsymbol{U}$ is unimodular, then $\boldsymbol{U}^{-1}$ is unimodular.*

Proof. We have $\det(\boldsymbol{U}^{-1}) = \frac{1}{\det(\boldsymbol{U})} \in \{-1, 1\}$. So, it remains to argue that $\boldsymbol{U}^{-1}$ has only integral entries. Set $\boldsymbol{U}^{ij} \in \mathbb{Z}^{n\times n}$ as the matrix where the ith column is replaced by the jth unit vector $\boldsymbol{e}_j$[3]. Then $\det(\boldsymbol{U}^{ij}) \in \mathbb{Z}$ and by *Cramer's rule*

$$U^{-1}_{ij} = \frac{\det(\boldsymbol{U}^{ij})}{\det(\boldsymbol{U})} \in \mathbb{Z}.$$

□

We will now see that two matrices span the same lattice if and only if they are they differ by a unimodular matrix:

Lemma 1.2. *Let $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathbb{R}^{n\times n}$ non-singular. Then $\Lambda(\boldsymbol{B}_1) = \Lambda(\boldsymbol{B}_2)$ if and only if there is a unimodular matrix $\boldsymbol{U}$ with $\boldsymbol{B}_2 = \boldsymbol{B}_1\boldsymbol{U}$.*

Proof. First, suppose that $\boldsymbol{B}_2 = \boldsymbol{B}_1\boldsymbol{U}$ with $\boldsymbol{U}$ being unimodular. The important observation is that the map $f : \mathbb{Z}^n \to \mathbb{Z}^n$ with $f(\boldsymbol{x}) := \boldsymbol{U}\boldsymbol{x}$ is a *bijection* on the integer lattice as $\boldsymbol{U}\boldsymbol{x} \in \mathbb{Z}^n$ for $\boldsymbol{x} \in \mathbb{Z}^n$ and any vector $\boldsymbol{y} \in \mathbb{Z}^n$ is hit by $\boldsymbol{U}(\boldsymbol{U}^{-1}\boldsymbol{y}) = \boldsymbol{y}$. Then

$$\Lambda(\boldsymbol{B}_2) = \{\boldsymbol{B}_2\boldsymbol{\lambda} \mid \boldsymbol{\lambda} \in \mathbb{Z}^n\} = \{\boldsymbol{B}_1\boldsymbol{U}\boldsymbol{\lambda} \mid \boldsymbol{\lambda} \in \mathbb{Z}^n\} = \Lambda(\boldsymbol{B}_1).$$

Now, let us go the other way around and assume that $\Lambda(\boldsymbol{B}_1) = \Lambda(\boldsymbol{B}_2)$. Then any column of $\boldsymbol{B}_1$ is an integral combination of columns in $\boldsymbol{B}_2$ and vice versa. We can use those integral coefficients to fill matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{Z}^{n\times n}$ so that $\boldsymbol{B}_2 = \boldsymbol{B}_1\boldsymbol{U}$ and $\boldsymbol{B}_1 = \boldsymbol{B}_2\boldsymbol{V}$. Then

$$\det(\boldsymbol{B}_1) = \det(\boldsymbol{B}_2\boldsymbol{V}) = \det(\boldsymbol{B}_1\boldsymbol{U}\boldsymbol{V}) = \det(\boldsymbol{B}_1) \cdot \det(\boldsymbol{U}) \cdot \det(\boldsymbol{V})$$

As $\det(\boldsymbol{U}), \det(\boldsymbol{V}) \in \mathbb{Z}$, we must have $\det(\boldsymbol{U}) \in \{-1, 1\}$ (and in fact it is not hard to argue that $\boldsymbol{V} = \boldsymbol{U}^{-1}$). □

Note that the unimodular matrix $\boldsymbol{U}$ can be found in polynomial time using row reduction / Gauss elimination. Hence, for given matrices $\boldsymbol{B}_1, \boldsymbol{B}_2$ one can test in polynomial time whether they generate the same lattice.

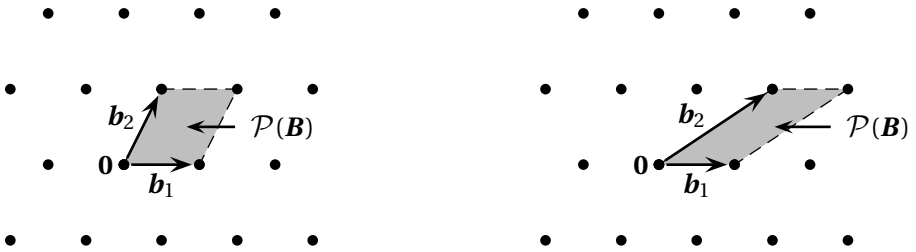
[3] Sounds like we have accidentally switched row and column indices — but it was on purpose.

1.1.2 The fundamental parallelepiped

The *fundamental parallelepiped* of the lattice $\Lambda(\boldsymbol{B})$ is the polytope

$$\mathcal{P}(\boldsymbol{B}) := \left\{ \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i \mid 0 \le \lambda_i < 1 \; \forall i \in [n] \right\}$$

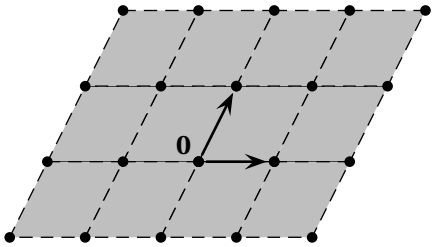
We see that this definition actually depends on the basis:



Let us make some observation: Since $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is a basis of $\mathbb{R}^n$, we know that for every $\boldsymbol{x} \in \mathbb{R}^n$ there is a unique coefficient vector $\boldsymbol{\lambda} \in \mathbb{R}^n$ so that $\boldsymbol{x} = \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i$. That means $\boldsymbol{x}$ can be written as

$$\boldsymbol{x} = \underbrace{\sum_{i=1}^{n} \lfloor \lambda_i \rfloor \boldsymbol{b}_i}_{\in \Lambda(\boldsymbol{B})} + \underbrace{\sum_{i=1}^{n} (\lambda_i - \lfloor \lambda_i \rfloor) \boldsymbol{b}_i}_{\in \mathcal{P}(\boldsymbol{B})}.$$

In other words, the *translates* of the parallelepiped placed at lattices points exactly partition the $\mathbb{R}^n$. We call this a *tiling* of $\mathbb{R}^n$. The tiling of the space with parallelepipeds can be visualized as follows:



Note that actually we can rewrite the fundamental parallelepiped as

$$\mathcal{P}(\boldsymbol{B}) = \{\boldsymbol{B}\boldsymbol{x} : \boldsymbol{x} \in [0,1[^n\},$$

that means it is the image of the hypercube $[0,1]^n$ under the linear map given by the matrix $\boldsymbol{B}$. Then by the *transformation formula*

$$\mathrm{vol}(\mathcal{P}(\boldsymbol{B})) = \underbrace{\mathrm{vol}([0,1[^n)}_{=1} \cdot |\det(\boldsymbol{B})|$$

While the fundamental parallelepiped itself does depend on the choice of the basis — its volume does not!

Lemma 1.3. *Let $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ and let $\Lambda := \Lambda(\boldsymbol{B})$ be the generated lattice. Then the* determinant of the lattice $\det(\Lambda) := |\det(\boldsymbol{B})|$ *is independent of the chosen basis. Moreover,* $\det(\Lambda) = \mathit{vol}(\mathcal{P}(\boldsymbol{B}))$.

Proof. Clear, because for different basis $\boldsymbol{B}, \boldsymbol{B}' \in \mathbb{R}^n$ of the same lattice, there is a unimodular transformation $\boldsymbol{U} \in \mathbb{Z}^{n\times n}$ with $\boldsymbol{B}' = \boldsymbol{B}\boldsymbol{U}$. Then $|\det(\boldsymbol{B}')| = |\det(\boldsymbol{B})| \cdot |\det(\boldsymbol{U})| = |\det(\boldsymbol{B})|$. □

Note that the quantity $\frac{1}{\mathrm{vol}(\mathcal{P}(\boldsymbol{B}))}$ gives the *density* of the lattice. For example if $\mathcal{B}(\mathbf{0}, R) := \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\|_2 \leq R\}$ is the ball of radius R around the origin, then

$$|\Lambda(\boldsymbol{B}) \cap \mathcal{B}(\mathbf{0}, R)| \approx \frac{\mathrm{vol}(\mathcal{B}(\mathbf{0}, R))}{\mathrm{vol}(\mathcal{P}(\boldsymbol{B}))}$$

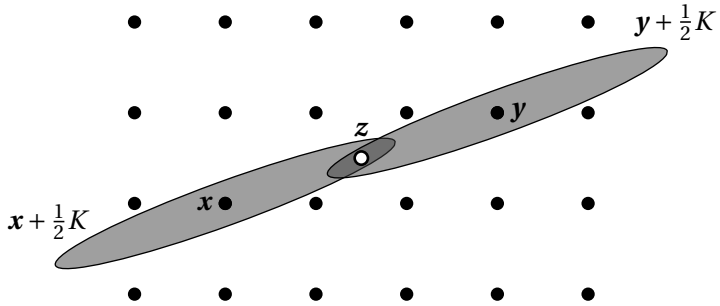
The "≈" should be understood that the ratio of both sides goes to 1 as $R \to \infty$. This is another more geometric argument why the volume of the fundamental parallelepiped cannot depend on the basis.

1.2 Minkowski's Theorem

Recall that a set $K \subseteq \mathbb{R}^n$ is called *convex* if for all $\boldsymbol{x}, \boldsymbol{y} \in K$ and $0 \leq \lambda \leq 1$ one also has $\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in K$. A set K is *(centrally) symmetric* if $\boldsymbol{x} \in K$ if and only if $-\boldsymbol{x} \in K$. In particular for a convex symmetric set K, we can define a norm that is called the *Minkowski norm* $\|\boldsymbol{x}\|_K := \min\{\lambda \geq 0 : \boldsymbol{x} \in \lambda K\}$. Here $\lambda K := \{\lambda\boldsymbol{x} \mid \boldsymbol{x} \in K\}$ is the scaled set. In other words, $\|\boldsymbol{x}\|_K$ gives the scaling factor that one needs until the scaled copy of K includes $\boldsymbol{x}$. For a vector $\boldsymbol{y} \in \mathbb{R}^n$, we define $\boldsymbol{y} + K := \{\boldsymbol{x} + \boldsymbol{y} \mid \boldsymbol{x} \in K\}$ as the *translate* of K by $\boldsymbol{y}$. Intuitively, $\|\boldsymbol{x}\|_K$ denotes by how much we need to scale K so that $\boldsymbol{x}$ is included. For example the Euclidean norm $\|\cdot\|_2$ is the norm $\|\cdot\|_K$ for $K := \mathcal{B}(\mathbf{0}, 1)$. The other way around, we defined a norm so that K is the *unit ball* of that norm and $\|\boldsymbol{x}\|_K \leq 1 \Leftrightarrow \boldsymbol{x} \in K$. We now come to Minkowski's Theorem which says that every large enough symmetric convex set must contain a non-zero lattice point.

Theorem 1.4 (Minkowski)**.** *Let $K \subseteq \mathbb{R}^n$ be a bounded symmetric convex set with vol$(K) > 2^n$. Then $K \cap (\mathbb{Z}^n \setminus \{\mathbf{0}\}) \neq \emptyset$.*

Proof. First, by assumption we have $\mathrm{vol}(\frac{1}{2}K) > 1$. Next, place copies of $\frac{1}{2}K$ at every lattice point in $\mathbb{Z}^n$.



Then on average, points in $\mathbb{R}^n$ are covered more than once. So there must be two different lattice points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^n$ so that the translates $\boldsymbol{x} + \frac{1}{2}K$ and $\boldsymbol{y} + \frac{1}{2}K$ overlapp. Let $\boldsymbol{z} \in (\boldsymbol{x} + \frac{1}{2}K) \cap (\boldsymbol{y} + \frac{1}{2}K)$ be a point in that intersection. Then

$$\|\boldsymbol{x} - \boldsymbol{y}\|_K \leq \underbrace{\|\boldsymbol{x} - \boldsymbol{z}\|_K}_{\leq 1/2} + \underbrace{\|\boldsymbol{y} - \boldsymbol{z}\|_K}_{\leq 1/2} \leq 1$$

Hence $(\boldsymbol{x} - \boldsymbol{y}) \in \mathbb{Z}^n / \{\mathbf{0}\}$ is the lattice point that we are looking for. □

Admittedly, this argument was a bit informal as we talked about the average density of an infinite covering. But one can make the argument nicely finite. Let $[-D,D]^n \supseteq K$ be a bounding box with $D \in \mathbb{N}$ containing our bounded convex set. Then for $R \in \mathbb{N}$, the box $[-(R+D),(R-D)]^n$ fully contains at least $(2R+1)^n$ translates. Hence we have

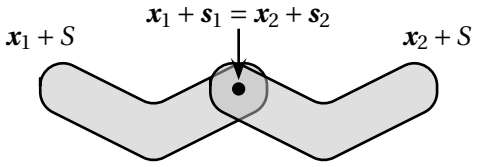
$$\frac{\text{vol}(\text{translates in } [-(R+D),R+D]^n)}{\text{vol}([-(R+D),(R+D)]^n)} \geq \frac{(2R+1)^n \cdot \text{vol}(\frac{1}{2}K)}{(2R+2D)^n} > 1$$

if we choose R large enough.

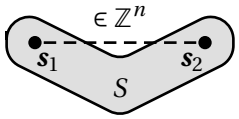
There is another theorem that is closely related to the one of Minkowski. Suppose that S is an arbitrary set; then S could be large without containing an integer point. But it still has to contain *differences* that are integral.

Theorem 1.5 (Blichfeldt)**.** *Let $S \subseteq \mathbb{R}^n$ be a measurable set with vol$(S) > 1$. Then there are $\boldsymbol{s}_1, \boldsymbol{s}_2 \in S$ with $\boldsymbol{s}_1 - \boldsymbol{s}_2 \in \mathbb{Z}^n$.*

Proof. Again place copies of $\boldsymbol{x} + S = \{\boldsymbol{x} + \boldsymbol{s} \mid \boldsymbol{s} \in S\}$ for all $\boldsymbol{x} \in \mathbb{Z}^n$. We will skip the usual compactness argument here, but there will be different points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{Z}^n$ with $(\boldsymbol{x}_1 + S) \cap (\boldsymbol{x}_2 + S) \neq \emptyset$.



Let $\boldsymbol{s}_1, \boldsymbol{s}_2 \in S$ be the points with $\boldsymbol{x}_1 + \boldsymbol{s}_1 = \boldsymbol{x}_2 + \boldsymbol{s}_2$. Rearranging gives $\boldsymbol{s}_1 - \boldsymbol{s}_2 = \boldsymbol{x}_2 - \boldsymbol{x}_1 \in \mathbb{Z}^n$. This gives the claim.



□

1.2.1 Minkowski's Theorem for general lattices

It is not hard to extend Minkowski's Theorem from the integer lattice $\mathbb{Z}^n$ to an arbitrary lattice. The idea is that any lattice $\Lambda(\boldsymbol{B})$ is the image of the integer lattice under the map $\boldsymbol{x} \mapsto \boldsymbol{B}\boldsymbol{x}$.

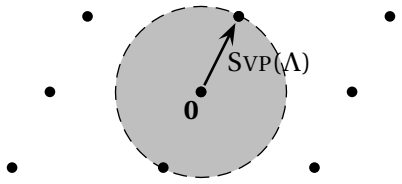
Theorem 1.6 (Minkowski's First Theorem)**.** *Let Λ be a lattice and K be a symmetric convex set with vol$(K) > 2^n \det(\Lambda)$. Then $K \cap (\Lambda \setminus \{\boldsymbol{0}\}) \neq \emptyset$.*

Proof. Define a bijective map $f : \mathbb{R}^n \to \mathbb{R}^n$ by $f(\boldsymbol{x}) = (x_1\boldsymbol{b}_1, \ldots, x_n\boldsymbol{b}_n)$. Then $f^{-1}(K) = \{\boldsymbol{\lambda} \in \mathbb{R}^n \mid B\boldsymbol{\lambda} \in K\}$. It is not hard to see that also $f^{-1}(K)$ is symmetric and convex. So the goal is to prove that $f^{-1}(K)$ contains an integral point. Using the transformation formula we know that $\text{vol}(f^{-1}(K)) = |\det(\boldsymbol{B}^{-1})| \cdot \text{vol}(K) = \frac{1}{\det(\Lambda)}\text{vol}(K) > 2^n$. Hence there is a non-trivial integral point $\boldsymbol{\lambda} \in f^{-1}(K)$ and $\boldsymbol{B}\boldsymbol{\lambda} \in K$ is the lattice point we are looking for. □

Note that if the set K is closed, then for compactness reasons it suffices if $\text{vol}(K) \geq 2^n \det(\Lambda)$ to have a non-zero lattice point. It should be clear that also Blichfeldt's Theorem works for arbitrary lattices (with the condition that $\text{vol}(S) > \det(\Lambda)$.

1.2.2 Minkowski's Theorem and the Shortest Vector

A particularly interesting vector in a lattice is the *Shortest Vector* with respect to the $\|\cdot\|_2$-norm. Let us abbreviate $\mathrm{SVP}(\Lambda) := \min\{\|\boldsymbol{x}\|_2 \mid \boldsymbol{x} \in \Lambda/\{\mathbf{0}\}\}$ as its length.



In fact, finding the shortest vector (or its length) is an **NP**-hard problem. However, one can get some estimates on it:

Theorem 1.7. *Any lattice $\Lambda \subseteq \mathbb{R}^n$ one has $\mathrm{SVP}(\Lambda) \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}$.*

Proof. First, for $r := \det(\Lambda)^{1/n}$, the hypercube $[-r,r]^n$ has a volume of

$$\mathrm{vol}([-r,r]^n) = (2r)^n \geq 2^n \det(\Lambda).$$

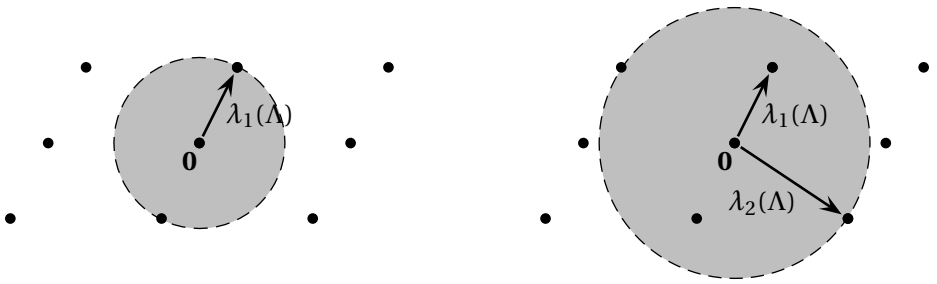
Hence there is a point $\boldsymbol{x} \in \Lambda/\{\mathbf{0}\}$ with $\|\boldsymbol{x}\|_\infty \leq \det(\Lambda)^{1/n}$. Of course $\|\boldsymbol{x}\|_2 \leq \sqrt{n} \cdot \|\boldsymbol{x}\|_\infty$, which implies the claim. □

Note that the scaling in the claim actually makes sense: if we scale the lattice Λ by a factor $s > 0$, then the length of the shortest vector also scales with s, while $\det(\Lambda)$ changes by a factor of s^n. The analysis can be improved to $c\sqrt{n}\det(\Lambda)^{1/n}$ for a constant $c < 1$ — however, this is best possible. There are lattices for every n with determinant 1 and shortest vector of length $\Omega(\sqrt{n})$.

In the literature, the length of the shortest vector is also written as $\lambda_1(\Lambda)$. This makes sense as more generally one can define

$$\lambda_i(\Lambda) := \min\{r \geq 0 \mid \dim(\mathrm{span}(\mathcal{B}(\mathbf{0},r) \cap \Lambda)) \geq i\}$$

as the *ith successive minimum*. That means one has i many linearly independent vectors of length at most λ_i and $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.



It turns out that one has at least a little bit of control over the lengths $\lambda_i(\Lambda)$:

Theorem 1.8 (Minkowski's Second Theorem)**.** *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$ one has*

$$\Big(\prod_{i=1}^{n} \lambda_i(\Lambda)\Big)^{1/n} \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}.$$

While Minkowski's First Theorem only tells us that $\lambda_i(\Lambda)$ is at most $\sqrt{n} \cdot \det(\Lambda)^{1/n}$, the Second Theorem gives the stronger statement that even the *geometric average* of $\lambda_1(\Lambda), \ldots, \Lambda_n(\Lambda)$ is bounded by that same quantity. But for the proof, we need some more ingredients. So we will give it at the end of this chapter.

1.2.3 Dirichlet's Theorem

We will now see another elegant application of Minkowski's First Theorem. Suppose we have a vector $\boldsymbol{\alpha} \in [0,1]^n$ of *real* numbers and we want to approximate the vector as good as possible with a vector of *rational* numbers so that the common denominator is at most a parameter Q. One can find some obvious applications in computer science, where one simply cannot work with real numbers but has to use rational approximations all the time. Then the most obvious choice would be

$$\left(\frac{\lceil \alpha_1 Q \rfloor}{Q}, \ldots, \frac{\lceil \alpha_n Q \rfloor}{Q}\right)$$

where $\lceil \cdot \rfloor$ rounds up or down to the nearest integer. One can easily see that the rounding error in every component is upper bounded by $\frac{1}{2Q}$. Is this best possible? Well, the task was to have a common denominator that is *at most* Q. So, we are allowed to pick any denominator in $\{1, \ldots, Q\}$, but we haven't made use of that freedom.

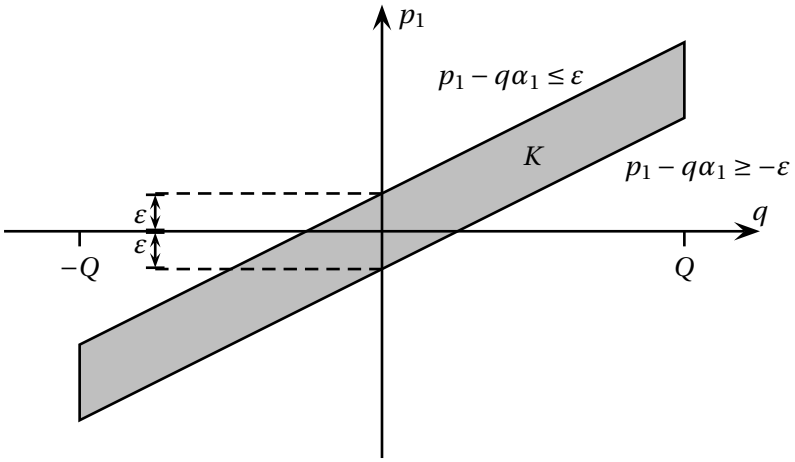
Theorem 1.9 (Dirichlet)**.** *For any $\boldsymbol{\alpha} \in]0,1]^n$ and $Q \in \mathbb{N}$, there are numbers $p_1, \ldots, p_n \in \mathbb{Z}_{\geq 0}$ and $q \in \{1, \ldots, Q\}$ so that*

$$\max_{i=1,\ldots,n} \left| \frac{p_i}{q} - \alpha_i \right| \leq \frac{1}{Q^{1/n} q}$$

Proof. If we abbreviate $\varepsilon := \frac{1}{Q^{1/n}}$ and multiply the inequality in the claim by q, then we get the constraint $|p_i - q \cdot \alpha_i| \leq \varepsilon$ for all $i = 1, \ldots, n$. Note that this is a linear constraint, hence we can reduce our problem to finding an integer point in the polytope

$$K := \left\{ (p_1, \ldots, p_n, q) \in \mathbb{R}^{n+1} \mid |p_i - q \cdot \alpha_i| \leq \varepsilon \; \forall i = 1, \ldots, n; \; |q| \leq Q \right\}$$

Note that on purpose, we admitted negative numbers to make the set K symmetric. For example for $n = 1$, one obtains the following picture:



One should think about K as a thin, but long "stab" along the line defined by $\boldsymbol{p} - q \cdot \boldsymbol{\alpha} = \mathbf{0}$. Geometrically speaking, the set K is a *parallelepiped* and his volume is equal to the volume of the box with length $2Q$ in one direction and 2ε in n directions. Hence

$$\mathrm{vol}(K) = 2Q \cdot (2Q^{-1/n})^n = 2^{n+1}.$$

Now we can apply Minkowski's theorem and we obtain $(p_1, \ldots, p_n, q) \in (K \cap \mathbb{Z}^{n+1}) \setminus \{\mathbf{0}\}$. For symmetry reasons, we can assume that $q \geq 0$. Note that it is impossible that $q = 0$, because otherwise $|p_i| \leq Q^{-1/n} < 1$ which implies that $p_1 = \ldots = p_n = 0$ and we would get a contradiction. Hence $q \in \{1, \ldots, Q\}$ and we have the desired approximation. □

1.3 The Gram Schmidt orthogonalisation

We have already mentioned that it is **NP**-hard to find the shortest vector in a lattice. Our goal is to be at least able to find an *approximate* shortest vector. That means for a lattice $\Lambda(\boldsymbol{B})$ we want to find a vector $\boldsymbol{x} \in \Lambda(\boldsymbol{B}) \setminus \{\mathbf{0}\}$ in polynomial time that has length $\|\boldsymbol{x}\|_2 \leq \alpha \cdot \mathrm{SVP}(\Lambda(\boldsymbol{B}))$. Here, $\alpha := \alpha(n) \geq 1$ is the so-called *approximation factor* that we would like to be as small as possible. Before we come to that algorithm, we need a useful procedure.

The *Gram-Schmidt orthogonalisation* takes linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^n$ as input and it computes an orthogonal basis $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ so that $\mathrm{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k) = \mathrm{span}(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_k^*)$ for all $k = 1, \ldots, n$. The idea is that we go through the vectors $\boldsymbol{b}_1, \ldots \boldsymbol{b}_n$ in that order and for each i we subtract all components of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$ from $\boldsymbol{b}_i$ and call the remainder $\boldsymbol{b}_i^*$. Formally the method is as follows:

Gram-Schmidt orthogonalisation

Input: Vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^n$

Output: Orthogonal basis $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$

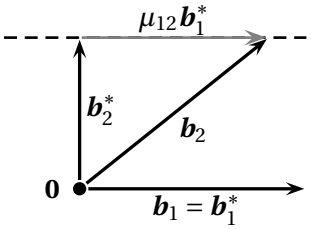
(1) $\boldsymbol{b}_1^* := \boldsymbol{b}_1$

(2) $\boldsymbol{b}_2^* := \boldsymbol{b}_2 - \mu_{1,2}\boldsymbol{b}_1^*$ with $\mu_{1,2} := \frac{\langle \boldsymbol{b}_2, \boldsymbol{b}_1^* \rangle}{\|\boldsymbol{b}_1^*\|_2^2}$

(3) ...

(j) $\boldsymbol{b}_j^* := \boldsymbol{b}_j - \sum_{i<j} \mu_{i,j}\boldsymbol{b}_i^*$ with $\mu_{ij} := \frac{\langle \boldsymbol{b}_j, \boldsymbol{b}_i^* \rangle}{\|\boldsymbol{b}_i^*\|_2^2} \quad \forall j = 1, \ldots, n$

Note that $\boldsymbol{b}_i^*$ is the *projection* of $\boldsymbol{b}_i$ on $\mathrm{span}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}\}^\perp$. For example for $n = 2$ the method can be visualized as follows:



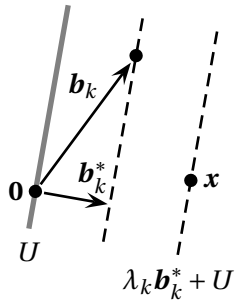
Note that the outcome of the Gram-Schmidt orthogonalization crucially depends on the *order* of the vectors. The next observation is that by *Cavalieri's principle*, the "shifting" does not change the volume of the fundamental parallelepiped. Hence

$$\det(\Lambda(\boldsymbol{B})) = \mathrm{vol}(\mathcal{P}(\boldsymbol{B})) = \prod_{i=1}^{n} \|\boldsymbol{b}_i^*\|_2.$$

The Gram-Schmidt orthogonalization gives us a nice lower bound on the length of a shortest vector.

Theorem 1.10. *Let $\boldsymbol{B}$ be a basis and $\boldsymbol{B}^* = (\boldsymbol{b}_1^*, \dots, \boldsymbol{b}_n^*)$ be its Gram-Schmidt orthogonalization. Then* $\text{SVP}(\Lambda(\boldsymbol{B})) \geq \min_{i=1,\dots,n} \|\boldsymbol{b}_i^*\|_2$.

Proof. Let $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ be any lattice vector and let $\boldsymbol{x} = \sum_{i=1}^n \lambda_i \boldsymbol{b}_i$ with $\lambda_i \in \mathbb{Z}$ be the linear combination that generates it. Let k be the largest index with $\lambda_k \neq 0$. Define the subspace $U := \text{span}\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_{k-1}\} = \text{span}\{\boldsymbol{b}_1^*, \dots, \boldsymbol{b}_{k-1}^*\}$.



Then $\boldsymbol{x}$ lies on a translate of that subspace which is $\lambda_k \boldsymbol{b}_k^* + U$. Hence $\|\boldsymbol{x}\|_2 \geq d(\boldsymbol{x}, U) = |\lambda_k| \cdot |\boldsymbol{b}_k^*|$ where $d(\boldsymbol{x}, U)$ tells the distance of $\boldsymbol{x}$ to U. □

In particular $\boldsymbol{b}_1^* = \boldsymbol{b}_1$ is always lattice vector — $\boldsymbol{b}_2^*, \dots, \boldsymbol{b}_n^*$ generally not. One idea to find a short lattice vector would be to find a basis (and an ordering on the vectors!) so that $\|\boldsymbol{b}_1^*\|_2 \leq \rho \cdot \|\boldsymbol{b}_i^*\|_2$ for all i and some ρ. Then by the previous lemma $\text{SVP}(\Lambda) \geq \min_{i=1,\dots,n} \|\boldsymbol{b}_i^*\|_2 \geq \frac{1}{\rho}\|\boldsymbol{b}_1\|_2$, hence $\boldsymbol{b}_1$ would be a ρ-approximation to the shortest vector. That is precisely what the LLL-algorithm will achieve.

1.4 The LLL-algorithm

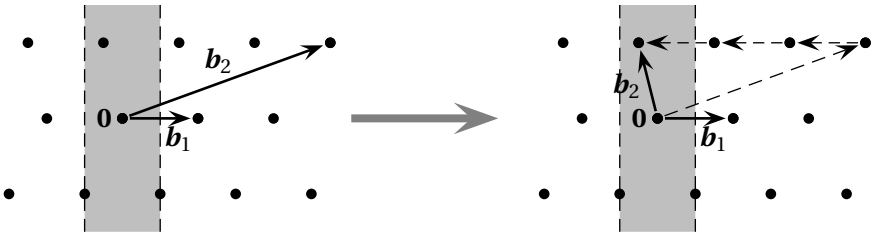
For the presentation of the LLL-algorithm, we are loosely following the exposition of Chris Peikert's excellent lecture notes [Pei13]. The main statement will be:

Theorem 1.11 (Lenstra-Lenstra-Lovász 1982)**.** *Given a regular matrix $\boldsymbol{B} \in \mathbb{Q}^{n \times n}$ one can compute a vector $\boldsymbol{x} \in \Lambda(\boldsymbol{B}) / \{\boldsymbol{0}\}$ of length $\|\boldsymbol{x}\|_2 \leq 2^{n/2} \cdot \text{SVP}(\Lambda(\boldsymbol{B}))$ in polynomial time.*

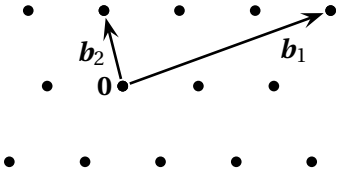
If $\boldsymbol{B}$ has integral entries then the running time is actually of the form $O(n^6 \log^3(n\|\boldsymbol{B}\|_\infty))$. The importance of this algorithm cannot be underestimated. Until now — 33 years after it's discovery — the LLL-algorithm is basically the only algorithm that gives any kind of non-trivial guarantee for any lattice problem in polynomial time!

Let us consider a basis $\boldsymbol{b}_1, \dots, \boldsymbol{b}_n$ (here for $n = 2$) and wonder what kind of operations we could do make the basis *as orthogonal as possible*, while it still generates the same lattice.

- *Subtracting vectors from each other:* In $n = 2$, if we have a vector $\boldsymbol{b}_2$, we can always subtract multiples of $\boldsymbol{b}_1$ from it so that $|\mu_{1,2}| \leq \frac{1}{2}$. In higher dimensions we will see that we can always achieve that $|\mu_{ij}| \leq \frac{1}{2}$ for all $i < j$.



- *Switching the order:* One the other hand, in $n = 2$ dimensions it might be that $\boldsymbol{b}_1$ is a lot longer than $\boldsymbol{b}_2$ so that we would not make progress in subtracting $\boldsymbol{b}_1$ from $\boldsymbol{b}_2$. But in that case we can swap the order of $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. In higher dimensions it will make sense to swap $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i+1}$ if $\|\boldsymbol{b}_i\|_2 \gg \|\boldsymbol{b}_{i+1}\|_2$.



1.4.1 Coefficient reduction

We want to begin by discussing how to make use of the first procedure, where we subtract integer multiples of vectors in the basis from other basis vectors. Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ be a basis and let $\mu_{ij} := \frac{\langle \boldsymbol{b}_j, \boldsymbol{b}_i^* \rangle}{\|\boldsymbol{b}_i^*\|_2^2}$ be the coefficients from the Gram-Schmidt orthogonalisation.

Definition 2. We call a basis *Coefficient-reduced* if $|\mu_{ij}| \leq \frac{1}{2}$ for all $1 \leq i < j \leq n$.

Lemma 1.12. *Given any basis $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ one can compute a coefficient-reduced basis $\tilde{\boldsymbol{B}}$ in polynomial time so that $\Lambda(\tilde{\boldsymbol{B}}) = \Lambda(\boldsymbol{B})$ and the Gram-Schmidt orthogonalizations are identical.*

Proof. Suppose that $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is a lattice basis and μ_{ij} are the Gram-Schmidt coefficients, that means

$$\boldsymbol{b}_j = \boldsymbol{b}_j^* + \sum_{i=1}^{j-1} \mu_{ij} \cdot \boldsymbol{b}_i^* \quad \forall\, j \in [n]. \qquad (*)$$

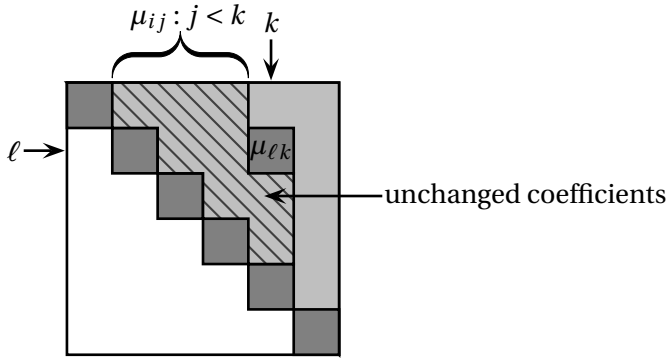
Now fix indices $1 \leq \ell < k \leq n$ and $q \in \mathbb{Z}$ and consider the updated basis $\tilde{\boldsymbol{B}}$ with vectors

$$\boldsymbol{b}_j := \begin{cases} \boldsymbol{b}_j + q \cdot \boldsymbol{b}_\ell & \text{if } j = k \\ \boldsymbol{b}_j & \text{otherwise.} \end{cases}$$

Clearly, $\Lambda(\tilde{\boldsymbol{B}}) = \Lambda(\boldsymbol{B})$. Let $\tilde{\mu}_{ij}$ be the updated Gram-Schmidt coefficients for $\tilde{\boldsymbol{B}}$. In particular $\tilde{\boldsymbol{b}}_1^* = \boldsymbol{b}_1^*, \ldots, \tilde{\boldsymbol{b}}_{k-1}^* = \boldsymbol{b}_{k-1}^*$ and for the coefficients we have $\tilde{\mu}_{ij} = \mu_{ij}$ for all pairs (i, j) with $i < j < k$ since only $\boldsymbol{b}_k$ has changed. Adding up $(*)$ for $j = k$ and $j = \ell$ we obtain

$$\tilde{\boldsymbol{b}}_k = \boldsymbol{b}_k + q \cdot \boldsymbol{b}_\ell = \boldsymbol{b}_k^* + \sum_{i<\ell} \underbrace{\left(\mu_{ik} + q \cdot \mu_{i\ell}\right)}_{=\tilde{\mu}_{ik}} \cdot \boldsymbol{b}_i^* + \underbrace{(\mu_{\ell k} + q)}_{=\tilde{\mu}_{\ell k}} \cdot \boldsymbol{b}_\ell^* + \sum_{i=\ell+1}^{k-1} \underbrace{\mu_{ik}}_{=\tilde{\mu}_{ik}} \boldsymbol{b}_i^*$$

Then we see that $\tilde{\mu}_{ik} = \mu_{ik}$ for all $i > \ell$. Moreover, we can choose $q \in \mathbb{Z}$ so that $|\tilde{\mu}_{\ell,k}| = |\mu_{\ell,k} + q| \leq \frac{1}{2}$. In the figure below we show which coefficients are guaranteed to not have changed:



Now suppose we denote the above procedure by $\tilde{\boldsymbol{B}} := \texttt{update}(\boldsymbol{B}, \ell, k)$. Then it is clear that we need to go through all index pairs (ℓ, k) in the right order and we can bring all coefficients into the interval $[-\frac{1}{2}, \frac{1}{2}]$:

(1) FOR $k = 1$ TO n DO

  (2) FOR $\ell = n$ DOWNTO $k+1$ DO

    (3) $\boldsymbol{B} := \texttt{update}(\boldsymbol{B}, \ell, k)$

That shows the claim. □

1.4.2 The main procedure

The crucial definition being the LLL-algorithm is the following:

Definition 3. Let $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ be a lattice basis and let μ_{ij} be the coefficients from the Gram-Schmidt orthogonalisation. The basis is called *LLL-reduced* if the following is satisfied

- *Coefficient reduced:* $|\mu_{i,j}| \leq \frac{1}{2}$ for all $1 \leq i < j \leq n$
- *Lovász condition:* $\|\boldsymbol{b}_i^*\|_2^2 \leq 2\|\boldsymbol{b}_{i+1}^*\|_2^2$ for $i = 1, \ldots, n-1$

First, let us see why this definition is desirable:

Lemma 1.13. *Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ be an LLL-reduced basis. Then $\|\boldsymbol{b}_1\|_2 \leq 2^{n/2} \cdot \textsc{Svp}(\Lambda(\boldsymbol{B}))$.*

Proof. Note that the 1st vector in the basis has $\boldsymbol{b}_1 = \boldsymbol{b}_1^*$. Applying the Lovász condition gives

$$\|\boldsymbol{b}_1\|_2^2 = \|\boldsymbol{b}_1^*\|_2^2 \leq 2\|\boldsymbol{b}_2^*\|_2^2 \leq \ldots \leq 2^{i-1} \cdot \|\boldsymbol{b}_i^*\|_2^2$$

On the other hand we can use Theorem 1.10 to lower bound the length of the shortest vector:

$$\textsc{Svp}(\Lambda(\boldsymbol{B}))^2 \geq \min_{i=1,\ldots,n} \|\boldsymbol{b}_i^*\|_2^2 \geq \min_{i=1,\ldots,n} 2^{-(i-1)} \|\boldsymbol{b}_1\|_2^2 \geq 2^{-n} \cdot \|\boldsymbol{b}_1\|_2^2$$

Taking square roots then gives the claim. □

This is the algorithm that will compute an LLL-reduced basis:

LLL-algorithm

Input: A lattice basis $\boldsymbol{B} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n) \in \mathbb{R}^{n \times n}$
Output: An LLL reduced basis $\tilde{\boldsymbol{B}} = (\tilde{\boldsymbol{b}}_1, \dots, \tilde{\boldsymbol{b}}_n)$

(1) Compute a Gram Schmidt orthogonalization $\boldsymbol{b}_1^*, \dots, \boldsymbol{b}_n^*$ with coefficients μ_{ij} and update whenever we change the order of the basis
(2) WHILE $\boldsymbol{B}$ is not LLL-reduced DO
  (3) Apply coefficient reduction so that $|\mu_{ij}| \leq \frac{1}{2}$
  (4) If there is an index i with $\|\boldsymbol{b}_i^*\|_2^2 > 2\|\boldsymbol{b}_{i+1}^*\|_2^2$ then swap $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i+1}$ in the ordering.

Obviously, the algorithm only terminates when it has found an LLL-reduced basis. Let us now prove that the algorithm terminates within a polynomial number of iterations. For this sake, we consider the *potential function*

$$\Phi(\boldsymbol{B}) = \prod_{i=1}^{n} \|\boldsymbol{b}_i^*\|_2^{n+1-i}$$

and we want to argue that it is decreasing. Intuitively, the potential function wants the vectors $\boldsymbol{b}_i^*$ with small i to be as small as possible. In particular, the point is that we swap i with $i+1$ if $\|\boldsymbol{b}_i^*\|_2$ is a lot longer than $\|\boldsymbol{b}_{i+1}^*\|_2$; one can expect that this should decrease the potential function. Let us define

$$\mathrm{vol}_k(\boldsymbol{b}_1, \dots, \boldsymbol{b}_k)$$

as the k-dimensional volume of the parallelepiped spanned by $\boldsymbol{b}_1, \dots, \boldsymbol{b}_k$. Note that by orthogonalizing the vectors, we do not change the volume of that parallelepiped (this is again *Cavalieri's principle*). Hence

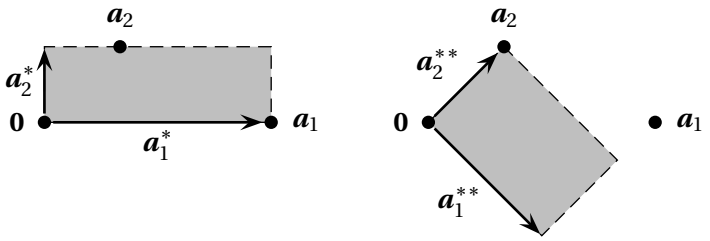
$$\mathrm{vol}_k(\boldsymbol{b}_1, \dots, \boldsymbol{b}_k) = \prod_{i=1}^{k} \|\boldsymbol{b}_i^*\|_2$$

We can use this to rewrite the potential function as

$$\Phi(\boldsymbol{B}) = \prod_{k=1}^{n} \mathrm{vol}_k(\boldsymbol{b}_1, \dots, \boldsymbol{b}_k)$$

We want to really understand what's going on, so here is a standalone lemma:

Lemma 1.14. *Suppose we have vectors $(\boldsymbol{a}_1, \boldsymbol{a}_2)$ with Gram Schmidt orthogonalization $(\boldsymbol{a}_1^*, \boldsymbol{a}_2^*)$ so that $\|\boldsymbol{a}_1^*\|_2^2 \geq 2\|\boldsymbol{a}_2^*\|_2^2$ and Let $\mu := \frac{\langle \boldsymbol{a}_1, \boldsymbol{a}_2 \rangle}{\|\boldsymbol{a}_1\|_2^2} \leq \frac{1}{2}$. Let $(\boldsymbol{a}_2^{**}, \boldsymbol{a}_1^{**})$ be the Gram Schmidt Orthogonalization for the reverse order $(\boldsymbol{a}_2, \boldsymbol{a}_1)$. Then $\|\boldsymbol{a}_2^{**}\|_2 \leq \sqrt{\frac{3}{4}} \cdot \|\boldsymbol{a}_1^*\|_2$.*



Proof. Let us first write down the vectors in both orthogonalizations depending on $\boldsymbol{a}_1, \boldsymbol{a}_2$ and the inner product μ:

$$\boldsymbol{a}_1^* = \boldsymbol{a}_1, \qquad \boldsymbol{a}_2^* = \boldsymbol{a}_2 - \underbrace{\frac{\langle \boldsymbol{a}_1, \boldsymbol{a}_2 \rangle}{\|\boldsymbol{a}_1\|_2^2}}_{=\mu} \boldsymbol{a}_1, \qquad \text{and} \qquad \boldsymbol{a}_2^{**} = \boldsymbol{a}_2$$

This can be rewritten to $\boldsymbol{a}_2^{**} = \boldsymbol{a}_2 = \boldsymbol{a}_2^* + \mu \boldsymbol{a}_1^*$. We inspect the square of the desired ratio and get:

$$\frac{\|\boldsymbol{a}_2^{**}\|_2^2}{\|\boldsymbol{a}_1^*\|_2^2} = \frac{\|\boldsymbol{a}_2^* + \mu \cdot \boldsymbol{a}_1^*\|_2^2}{\|\boldsymbol{a}_1^*\|_2^2} \overset{\text{Pythagoras \& } \boldsymbol{a}_1^* \perp \boldsymbol{a}_2^*}{=} \frac{\|\boldsymbol{a}_2^*\|_2^2 + \mu^2 \|\boldsymbol{a}_1^*\|_2^2}{\|\boldsymbol{a}_1^*\|_2^2} \leq \underbrace{\frac{\|\boldsymbol{a}_2^*\|_2^2}{\|\boldsymbol{a}_1^*\|_2^2}}_{\leq 1/2} + \frac{1}{4} \leq \frac{3}{4}$$

□

Lemma 1.15. *In each iteration $\Phi(\boldsymbol{B})$ reduces by a constant factor.*

Proof. First, observe that making $\boldsymbol{B}$ coefficient-reduced does not change the Gram-Schmidt orthogonalization and hence leaves the potential function $\Phi(\boldsymbol{B})$ unchanged. Now, let $\boldsymbol{B}$ be a coefficient-reduced basis at the beginning of (4) and let $\tilde{\boldsymbol{B}}$ be the basis after we swapped index i. Then the whole Gram-Schmidt orthogonalisation of $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$ is identical — except for vectors i and $i+1$. We can hence write

$$\frac{\Phi(\tilde{\boldsymbol{B}})}{\Phi(\boldsymbol{B})} = \frac{\mathrm{vol}_i(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}, \boldsymbol{b}_{i+1})}{\mathrm{vol}_i(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}, \boldsymbol{b}_i)}$$

Let $\boldsymbol{a}_1$ be the projection of $\boldsymbol{b}_i$ on the subspace $U := \mathrm{span}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}\}^\perp$ and let $\boldsymbol{a}_2$ be the projection of $\boldsymbol{b}_{i+1}$ on that subspace. In the notation of the previous lemma, let $(\boldsymbol{a}_1^*, \boldsymbol{a}_2^*)$ be the orthogonalization of $(\boldsymbol{a}_1, \boldsymbol{a}_2)$ (in that order) and let $(\boldsymbol{a}_2^*, \boldsymbol{a}_1^*)$ be the orthogonalization of $(\boldsymbol{a}_2, \boldsymbol{a}_1)$ (again in that order). Then $\boldsymbol{a}_1^* = \boldsymbol{b}_i^*$ and $\boldsymbol{a}_2^{**} = \tilde{\boldsymbol{b}}_{i+1}^*$. Since the volumes of both parallelepipeds is proportional to the distance of the last vector to the subspace U, we get

$$\frac{\Phi(\tilde{\boldsymbol{B}})}{\Phi(\boldsymbol{B})} = \frac{\mathrm{vol}_i(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}, \boldsymbol{b}_{i+1})}{\mathrm{vol}_i(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}, \boldsymbol{b}_i)} = \frac{\|\boldsymbol{a}_2^{**}\|_2}{\|\boldsymbol{a}_1^*\|_2} \leq \sqrt{\frac{3}{4}} < 1$$

using Lemma 1.14. □

Now it is easy to show that the LLL is a polynomial time algorithm. Originally, $\boldsymbol{B}$ could have been any matrix with rational entries. But we can scale any such matrix so that the entries become integral. Note that an entry B_{ij} used $\approx \log_2(|B_{ij}|)$ bits in the input. Hence, a polynomial time algorithm should have a running time that is polynomial in n and in $\log(\|\boldsymbol{B}\|_\infty)$. Moreover, the squared volume of a parallelepiped that is spanned by integral vectors will be integral.

Lemma 1.16. *Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k \in \mathbb{Z}^n$ be linearly independent integral vectors. Then $vol_k(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)^2 \in \mathbb{Z}_{\geq 1}$.*

Proof sketch. We have restricted our attention to full rank lattices so far. But there are more general definitions for lattices of lower rank. Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k) \in \mathbb{Z}^{n \times k}$. Consider the lattice $\Lambda(\boldsymbol{B})$ spanned by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$. One can define show that[4]

$$\mathrm{vol}_k(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k) = \sqrt{\det(\boldsymbol{B}^T \boldsymbol{B})}$$

Since $\boldsymbol{B}^T \boldsymbol{B} \in \mathbb{Z}^{k \times k}$ is an integral matrix one has $\det(\boldsymbol{B}^T \boldsymbol{B}) \in \mathbb{Z}$ and the claim follows. □

[4] The reader may want to double-check that in case of $k = n$, this is exactly $|\det(\boldsymbol{B})|$.

The last lemma implies that in particular $\mathrm{vol}_k(\boldsymbol{b}_1,\dots,\boldsymbol{b}_k) \geq 1$ for all $k = 1,\dots,n$.

Lemma 1.17. *Suppose that $\boldsymbol{B} \in \mathbb{Z}^{n\times n}$. Then the LLL-algorithm applied to $\boldsymbol{B}$ takes $O(n^2 \log\max\{n, \|\boldsymbol{B}\|_\infty\})$ many iterations.*

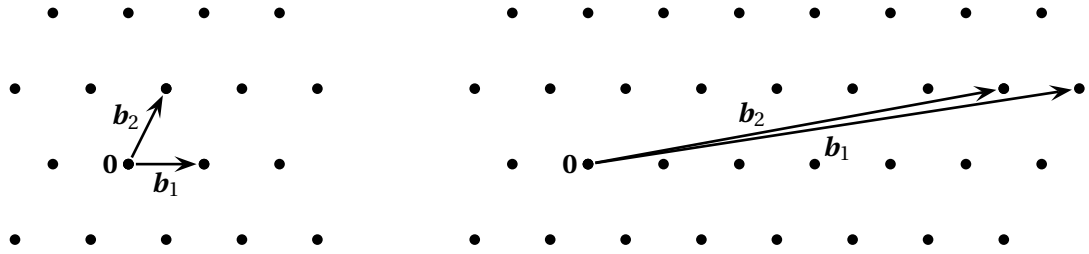
Proof. Suppose that $\boldsymbol{b}_1,\dots,\boldsymbol{b}_n$ are the original columns of the input matrix $\boldsymbol{B}$. Since the Gram-Schmidt process is a projection, it cannot make vectors longer. Hence $\|\boldsymbol{b}_i^*\|_2 \leq \|\boldsymbol{b}_i\|_2 \leq \sqrt{n}\cdot\|\boldsymbol{B}\|_\infty$. Hence before the first iteration, the potential function is bounded by $\Phi(\boldsymbol{B}) \leq (\sqrt{n}\cdot\|\boldsymbol{B}\|_\infty)^{n^2}$. Now, let $\tilde{\boldsymbol{B}} = (\tilde{\boldsymbol{b}}_1,\dots,\tilde{\boldsymbol{b}}_n)$ be a matrix in an arbitrary iteration of the LLL algorithm. Since $\tilde{\boldsymbol{B}}$ is obtained by subtracting and permuting columns in the integral matrix $\boldsymbol{B}$, we know that $\tilde{\boldsymbol{B}} \in \mathbb{Z}^{n\times n}$. As we observed earlier, we have

$$\Phi(\tilde{\boldsymbol{B}}) = \prod_{k=1}^{n} \underbrace{\mathrm{vol}_k(\boldsymbol{b}_1,\dots,\boldsymbol{b}_k)}_{\geq 1} \geq$$

Since the potential function decreases by a constant factor in each iteration, the claim follows. □

1.4.3 The orthogonality defect

We want to further discuss that the LLL-algorithm does not only find the shortest vector, but the LLL-reduced basis has a lot more properties. The LLL-reduced basis is really a "good" basis in the sense that at least it is approximately orthogonal.



"good" basis "bad" basis

For a lattice basis $\boldsymbol{B} = (\boldsymbol{b}_1,\dots,\boldsymbol{b}_n)$, we define the *orthogonality defect* as

$$\gamma(\boldsymbol{B}) := \frac{\prod_{i=1}^{n}\|\boldsymbol{b}_i\|_2}{\prod_{i=1}^{n}\|\boldsymbol{b}_i^*\|_2}$$

Note that $\gamma(\boldsymbol{B}) \geq 1$ and we have $\gamma(\boldsymbol{B}) = 1$ if and only if $\boldsymbol{b}_1,\dots,\boldsymbol{b}_n$ are pairwise orthogonal. So, $\gamma(\boldsymbol{B})$ is indeed a measure for how orthogonal a basis is. Even from a non-constructive viewpoint, it is non-trivial to argue that there even exists a basis with $\gamma(\boldsymbol{B})$ bounded by some function of n for any lattice.

Lemma 1.18. *The orthogonality defect of an LLL-reduced basis $\boldsymbol{B} = (\boldsymbol{b}_1,\dots,\boldsymbol{b}_n)$ is $\gamma(\boldsymbol{B}) \leq 2^{n^2/2}$.*

Proof. Now it is important that we also have control over intermediate vectors. Recall that by the properties of the Gram-Schmidt orthogonalization, we have

$$\boldsymbol{b}_k = \boldsymbol{b}_k^* + \sum_{i=1}^{k-1} \mu_{ik}\boldsymbol{b}_i^*.$$

Taking norms, we get

$$\|\boldsymbol{b}_k\|_2^2 = \left\|\boldsymbol{b}_k^* + \sum_{i=1}^{k-1} \mu_{ik}\boldsymbol{b}_i^*\right\|_2^2 = \|\boldsymbol{b}_k^*\|_2^2 + \sum_{i=1}^{k-1} \underbrace{\mu_{ik}^2}_{\leq 1/4} \underbrace{\|\boldsymbol{b}_i^*\|_2^2}_{\leq 2^{k-i}\|\boldsymbol{b}_k^*\|_2^2} \leq \|\boldsymbol{b}_k^*\|_2^2 \cdot \underbrace{\left(1 + \frac{1}{4}\sum_{i=1}^{k-1} 2^{k-i}\right)}_{\leq 2^k} \leq 2^k \cdot \|\boldsymbol{b}_k^*\|_2^2$$

using that $\|\boldsymbol{b}_j^*\|_2^2 \leq 2\|\boldsymbol{b}_{j+1}\|_2^2$ for all $j = 1, \ldots, n-1$. Then being generous with the constants, we get that

$$\gamma(\boldsymbol{B}) = \prod_{k=1}^{n} \underbrace{\frac{\|\boldsymbol{b}_k\|_2}{\|\boldsymbol{b}_k^*\|_2}}_{\leq 2^{n/2}} \leq 2^{n^2/2}.$$

□

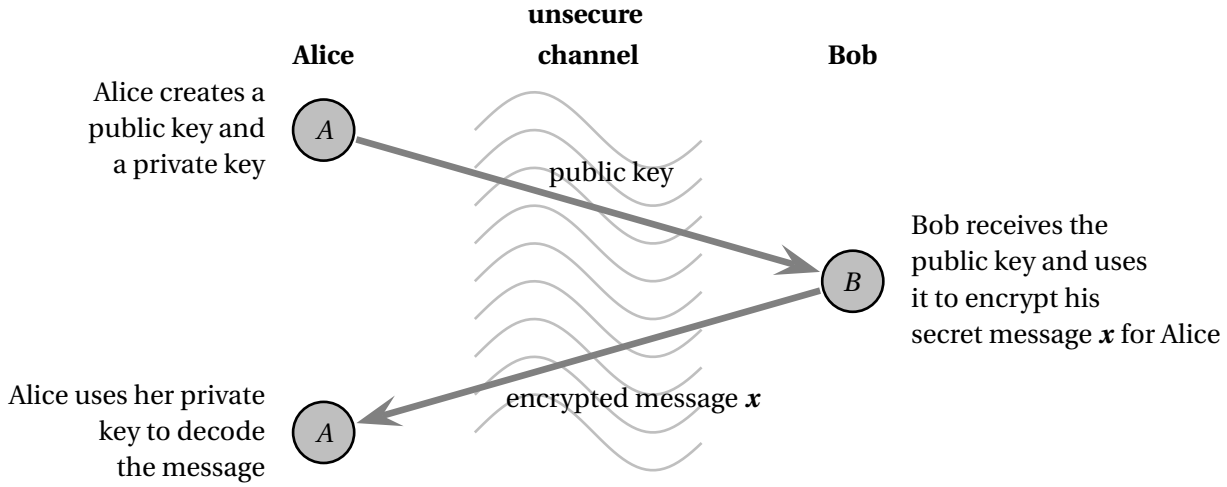
A different definition of a reduced basis is due to Korkhine and Zolotarav and is called *K-Z-reduced basis*. Such a basis has an orthogonality defect of at most n^n. However, no polynomial time algorithm is known to compute such a basis.

Open Question 1. Is there a polynomial time algorithm to find $\boldsymbol{x} \in \Lambda(\boldsymbol{B})/\{\boldsymbol{0}\}$ with $\|\boldsymbol{x}\|_2 \leq \text{poly}(n) \cdot \text{SVP}(\Lambda(\boldsymbol{B}))$? As we will outline later, the problem of approximation SVP within a factor of n is in **NP** ∩ **coNP** (and using more advanced arguments one can bring the factor down do $\sqrt{n}$). On the other hand, depending on the polynomial, such an algorithm would break existing lattice based cryptosystems.

1.5 Breaking Knapsack Cryptosystems

The approximation guarantee of $2^{n/2}$ provided by the LLL-algorithm may sound weak, but it is already enough for a couple of very surprising applications. We want to show-case one of them here.

For a *public key cryptosystem*, the goal is that two parties A (say Alice) and B (say, Bob) can communicate a secret message over a public channel without that any third party C could decrypt it.

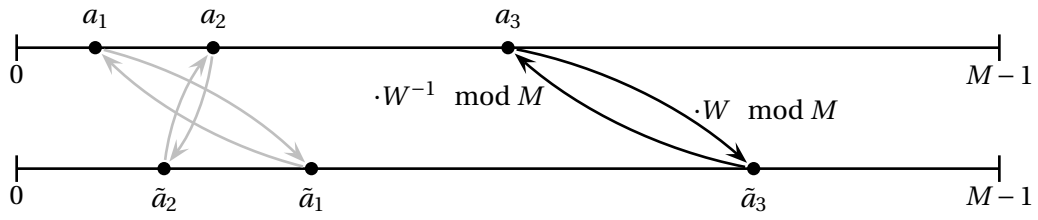


To be more precise, there will be a *public key*, which Alice would generate and then put on her webpage. Then Bob could see that public key and he would use it to encrypt a message that he wants to send to Alice. Even if a third party wiretaps the message that Bob sends to Alice and knows the public key, we want that C is still unable to decrypt the message. The important fact behind such a cryptosystem is that Alice and Bob do not need to agree on any key in advance – it suffices to communicate over the unsecure channel.

The idea behind the *Knapsack cryptosystem* is the following: Alice could create some large numbers $a_1, \dots, a_n \in \mathbb{N}$ and publish them as public key. If Bob wants to send a secret message $\boldsymbol{x} \in \{0,1\}^n$ to Alice, he could compute the sum $S := \sum_{i=1}^n a_i x_i$ and send Alice the number S. Knapsack is an **NP**-hard problem, so Bob could hope that his message is safe enough. But in any meaningful cryptosystem, at least the intended receiver Alice should be able to decrypt the message efficiently. So, the Knapsack instance has to be "simple". One way of having an easy Knapsack instance is if the numbers a_i are *super-increasing*, that means

$$a_i > \sum_{j<i} a_j \quad \forall i \in [n].$$

It is a simple exercise to show that in this case, given a sum $S = \sum_{i=1}^n a_i x_i$, one can recover the vector $\boldsymbol{x} \in \{0,1\}^n$ with a polynomial time greedy-style algorithm. But of course, also a third party C would know how to solve such a Knapsack problem. So, we need one more ingredient to "hide" the super increasing sequence. We simply take a large number $M > \sum_{i=1}^n a_i$ and some random $W \in \{1, \dots, M-1\}$ with $\gcd(M,W) = 1$ and compute numbers $\tilde{a}_i := a_i \cdot W \mod M$. Here it is helpful to remember that the map $a \mapsto a \cdot W \mod M$ is a bijection on $\{0, \dots, M-1\}$ and the inverse function is simply $a \mapsto a \cdot W^{-1} \mod M$. The inverse W^{-1} with $W \cdot W^{-1} \equiv_M 1$ exists as M and W are coprime. We use that function to randomly "shuffle" the numbers:



Now, those numbers $\tilde{a}_1, \dots, \tilde{a}_n$ form the public key. Note that the numbers of the super increasing sequence are now wildly mixed. Obviously Alice should randomly permute the indices, but to keep notation simple, we skip this detail. Now, Bob receives the public key $\tilde{a}_1, \dots, \tilde{a}_n$ and computes the sum $\tilde{S} = \sum_{i=1}^n \tilde{a}_i x_i$. Then he sends the message $\tilde{S}$ to Alice. Alice computes $S := \tilde{S} \cdot W^{-1} \mod M$ and then uses the super-increasing property to compute the unique vector $\boldsymbol{x}$ satisfying $S = \sum_{i=1}^n a_i x_i$ (here we use that $\sum_{i=1}^n a_i x_i \leq M$). Note that her *private key* consists of the pair (M,W) (and the permutation of the indices).

It seemed that in order to decrypt the message without knowing the private key (the numbers M, W and the order of the super-increasing sequence), one would have to solve the Knapsack problem. The remaining incredient for a working cryptosystem would be the generation of a hard Knapsack instance. Intuitively one might think that taking a random instance with large enough coefficients would give such a hard instance. Surprisingly, this is false due to the LLL algorithm.

1.5.1 A polynomial time algorithm to solve sparse knapsack instance

While Knapsack is **NP**-hard in the worst case, it turned out that very sparse Knapsack instances admit polynomial time algorithms. Note that here, we do not try to optimize any constant.

Theorem 1.19 (Lagarias, Odlyzko 1985)**.** *Suppose we generate a Knapsack instance by picking independently $a_1, \ldots, a_n \sim \{\frac{1}{2} \cdot 2^{4n^2}, \ldots, 2^{4n^2}\}$ at random. Then take a vector $\boldsymbol{x} \in \{0,1\}^n$ and compute $S := \sum_{i=1}^n a_i x_i$. Then there is a polynomial time algorithm which on input $(\boldsymbol{a}, S)$, with high probability recovers the vector $\boldsymbol{x}$.*

Proof. Let us define an $(n+1)$-dimensional basis

$$\boldsymbol{B} = (\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ 2^n S & -2^n a_1 & -2^n a_2 & -2^n a_3 & \ldots & -2^n a_n \end{pmatrix}$$

and let us consider the lattice $\Lambda(\boldsymbol{B})$ that is spanned by the columns $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_n$ of this matrix. First of all, we claim that this lattice contains a very short vector, namely $\boldsymbol{b}_0 + \sum_{i=1}^n x_i \boldsymbol{b}_i = \begin{pmatrix} \boldsymbol{x} \\ 0 \end{pmatrix}$. The length of this vector is $\|\boldsymbol{x}\|_2 \leq \sqrt{n}$. Hence, we can use the LLL-algorithm to compute a vector in the lattice $\Lambda(\boldsymbol{B})$ of length at most $\|\boldsymbol{x}\|_2 \cdot 2^{n/2} \ll 2^n$. It remains to show that any such short vector is actually a multiple of $(\boldsymbol{x}, 0)$ (in fact, the LLL-algorithm returns a basis and the following claim even implies that the shortest vector in that basis must be $\pm(\boldsymbol{x}, 0)$):

Claim: With high probability, the only vectors $\boldsymbol{z} \in \Lambda(\boldsymbol{B})$ with $\|\boldsymbol{z}\|_2 < 2^n$ are multiples of $(\boldsymbol{x}, 0)$.

Proof: Let $\boldsymbol{z} \in \Lambda(\boldsymbol{B})$ be a lattice vector with $\|\boldsymbol{z}\|_2 < 2^n$. We can write $\boldsymbol{z} = \alpha \boldsymbol{b}_0 + \sum_{i=1}^n y_i \boldsymbol{b}_i$ for integer coefficients $\alpha, y_1, \ldots, y_n \in \mathbb{Z}$. The last coordinate of $\boldsymbol{z}$ is $2^n \cdot (\alpha S - \sum_{i=1}^n y_i a_i)$. Since the absolute value of this number has to be less than 2^n, we know that indeed $\sum_{i=1}^n y_i a_i = \alpha S$. Since $\|\boldsymbol{z}\|_2 < 2^n$, we also know that $\|\boldsymbol{y}\|_\infty \leq 2^n$. Then $\alpha \leq 2n \cdot 2^n$, since otherwise, we would have $|\alpha| S > \sum_{i=1}^n a_i y_i$.

Note that the number of triples $(\boldsymbol{x}, \boldsymbol{y}, \alpha)$ with $\boldsymbol{x} \in \{0,1\}^n$, $\|\boldsymbol{y}\|_\infty \leq 2^n$ and $|\alpha| \leq 2n \cdot 2^n$ is bounded by $2n \cdot 2^{n \cdot (n+1)} \cdot 2^n \cdot 3^{2n+1} \leq 2^{3n^2}$. So, it suffices to show the following:

Claim: Fix a triple $(\boldsymbol{x}, \boldsymbol{y}, \alpha)$ with $\boldsymbol{y} \notin \mathbb{Z}\boldsymbol{x}$. Then

$$\Pr_{\boldsymbol{a} \sim [2^{4n^2}]^n} \left[\sum_{i=1}^n y_i a_i = \alpha \cdot \left(\sum_{i=1}^n a_i x_i \right) \right] \leq \frac{1}{2} \cdot 2^{-4n^2} \quad (*)$$

Proof: By assumption $\boldsymbol{y}$ is not a multiple of $\boldsymbol{x}$, so we know that so there is an index j with $y_j \neq \alpha x_j$. Now suppose that the value of $a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_n$ has been fixed and we just pick $a_j \sim \{\frac{1}{2} \cdot 2^{4n^2}, \ldots, 2^{4n^2}\}$ at random. Then the equation in $(*)$ can be rearranged to

$$a_j \underbrace{(y_j - \alpha x_j)}_{\neq 0} = \alpha \sum_{i \neq j} x_i a_i - \sum_{i \neq j} y_i a_i \quad (**)$$

It doesn't actually matter what the right hand side of $(**)$ is, just observe that there will be at most one choice of a_j that could satisfy the equation. The bound on the probability follows since we are drawing a_j from $\{\frac{1}{2} \cdot 2^{4n^2}, \ldots, 2^{4n^2}\}$. □

The original attack on the Knapsack cryptosystem was by Shamir. The generalized argument for low-density subset-sum problem is by Lagarias and Odlyzko with a later simplification of Frieze. For more information, we recommend the survey *The Rise and Fall of Knapsack Cryptosystems* by Odlyzo [Odl90].

1.6 The dual lattice and applications

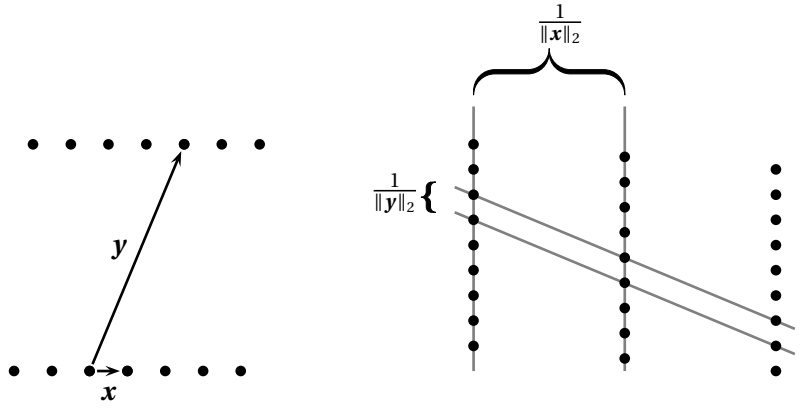
An important point is that Minkowski's Theorem is inherently *non-constructive*, that means given a symmetric convex set K with $\mathrm{vol}(K) > 2^n$, we know it must contain a non-zero integer point — but the proof method does not provide a polynomial time algorithm to find that point. One might think that this is a pure artefact of the proof and a different proof technique will be algorithmic. But such a proof would have tremendous consequences. The reader may be reminded, that there are cryptosystems that rely on the assumption that it is hard to approximate the Shortest Vector Problem up to small polynomial factors. Moreover, one can show that even an approximate constructive proof for Minkowski's Theorem would imply an approximation algorithm for SVP. The proof is via an application of the concept of dual lattices.

1.6.1 Dual lattices

Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice. Then the *dual lattice* is

$$\Lambda^* = \{\boldsymbol{y} \in \mathbb{R}^n \mid \langle \boldsymbol{y}, \boldsymbol{x} \rangle \in \mathbb{Z} \ \forall \boldsymbol{x} \in \Lambda\}$$

In order to get some intuition, take a lattice vector $\boldsymbol{x} \in \Lambda$. Then all dual lattice vectors lie on the hyperplanes $H_k = \{\boldsymbol{y} \in \mathbb{R}^n \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle = k\}$. In particular, each dual vector $\boldsymbol{y} \in \Lambda^*$ will either be orthogonal to $\boldsymbol{x}$ or it will have a separation of at least $\frac{1}{\|\boldsymbol{x}\|_2}$ from H_0. Loosely speaking, if lattice vectors in some direction are short in Λ, they will be long in this direction in Λ^* and vice versa. We reproduce a helpful picture from [Reg09]:



lattice Λ dual lattice Λ^*

The next lemma will show that Λ^* is indeed the lattice and we will see that it's basis is just the transpose inverse of the original basis:

Lemma 1.20. *Let $\Lambda := \Lambda(\boldsymbol{B})$ be a full rank lattice generated by $\boldsymbol{B} \in \mathbb{R}^{n \times n}$. Then $(\boldsymbol{B}^{-1})^T$ is a lattice basis for the dual lattice Λ^*.*

Proof. Let us define $\Lambda' := \{(\boldsymbol{B}^{-1})^T \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \mathbb{Z}^n\}$ as the candidate lattice. For any vectors $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{\lambda} \in \Lambda$ and $\boldsymbol{y} = (\boldsymbol{B}^{-1})^T \boldsymbol{\mu}$ with coefficients $\boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathbb{Z}^n$, the inner product is

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = (\boldsymbol{B}\boldsymbol{\lambda})^T ((\boldsymbol{B}^{-1})^T \boldsymbol{\mu}) = \boldsymbol{\lambda}^T \underbrace{\boldsymbol{B}^T (\boldsymbol{B}^{-1})^T}_{=\boldsymbol{I}} \boldsymbol{\mu} = \langle \boldsymbol{\lambda}, \boldsymbol{\mu} \rangle \in \mathbb{Z}$$

and hence $\Lambda' \subseteq \Lambda^*$.

For the other direction, take a point $\boldsymbol{y} \in \Lambda^*$. Since $(\boldsymbol{B}^{-1})^T$ has full rank, there must be a unique vector $\boldsymbol{\mu} \in \mathbb{R}^n$ with $\boldsymbol{y} = (\boldsymbol{B}^{-1})^T \boldsymbol{\mu}$. Then

$$\mathbb{Z} \overset{\text{Def. dual lattice}}{\supseteq} \{\langle \boldsymbol{x}, \boldsymbol{y} \rangle : \boldsymbol{x} \in \Lambda\} = \{\langle \boldsymbol{\lambda}, \boldsymbol{\mu} \rangle : \boldsymbol{\lambda} \in \mathbb{Z}^n\} \overset{\boldsymbol{\lambda} = \boldsymbol{e}_i}{\supseteq} \{\mu_1, \ldots, \mu_n\}$$

We see that $\boldsymbol{\mu} \in \mathbb{Z}^n$ and hence $\Lambda^* \subseteq \Lambda'$. □

We can use this insight to derive

Lemma 1.21. *For a lattice $\Lambda = \Lambda(\boldsymbol{B})$ the following holds:*

(i) $(\Lambda^*)^* = \Lambda$
(ii) $\det(\Lambda^*) = \frac{1}{\det(\Lambda)}$.

Proof. The first claim follows from the last lemma with the observation that $(((\boldsymbol{B}^{-1})^T)^{-1})^T = \boldsymbol{B}$. The 2nd claim follows from $\det(\Lambda^*) \cdot \det(\Lambda) = |\det((\boldsymbol{B}^{-1})^T) \cdot \det(\boldsymbol{B})| = |\det(\boldsymbol{B}^{-1}\boldsymbol{B})| = |\det(\boldsymbol{I})| = 1$. □

1.6.2 Solving Shortest Vector via Minkowski's Theorem

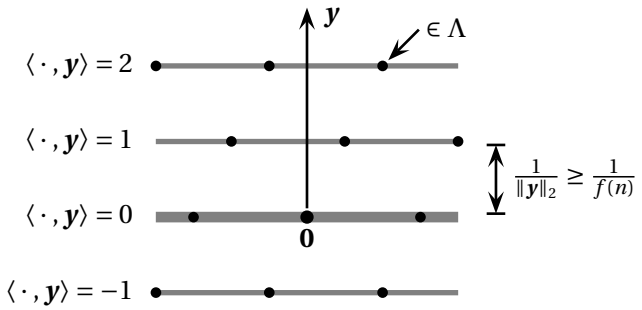
We will implicitly use the following lemma — we we postpone the proof for the moment.

Lemma 1.22. *Let $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ be a basis for lattice $\Lambda := \Lambda(\boldsymbol{B})$ and let $\boldsymbol{y} \in \mathbb{Q}^n$. Then there is a polynomial time algorithm to find a lattice basis $\boldsymbol{B}'$ for the sublattice $\Lambda' := \{\boldsymbol{x} \in \Lambda \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0\}$.*

Recall that Minkowski's Theorem guarantees that there is always a lattice vector $\boldsymbol{x}$ with $\|\boldsymbol{x}\|_2 \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}$. Now the proof that an approximate version of Minkowski's theorem would imply an approximation algorithm for the Shortest Vector Problem.

Theorem 1.23 (Lenstra-Schnorr 1990)**.** *Suppose that we have a polynomial time algorithm that for any lattice Λ is able to find a vector $\boldsymbol{x} \in \Lambda \setminus \{\boldsymbol{0}\}$ with $\|\boldsymbol{x}\|_2 \leq f(n) \cdot \det(\Lambda)^{1/n}$, where $f(n)$ is a non-decreasing function. Then there exists a polynomial time $f(n)^2$-approximation algorithm for* SVP.

Proof. For the sake of a simpler notation, let us rescale the lattice Λ so that $\det(\Lambda) = 1$ and also $\det(\Lambda^*) = 1$. First, we use the assumed algorithm to find a lattice vector $\boldsymbol{x} \in \Lambda \setminus \{\boldsymbol{0}\}$ of length $\|\boldsymbol{x}\|_2 \leq f(n)$. Is that point $\boldsymbol{x}$ a good approximation for SVP? Well, not always as it is perfectly possible that $\text{SVP}(\Lambda) \ll 1$. So, we apply the algorithm again, but now to find a vector $\boldsymbol{y} \in \Lambda^* / \{\boldsymbol{0}\}$ in the *dual lattice*, again with $\|\boldsymbol{y}\|_2 \leq f(n)$. Crucially, by definition of the dual lattice, we know that $\langle \boldsymbol{x}', \boldsymbol{y} \rangle \in \mathbb{Z}$ for all $\boldsymbol{x}' \in \Lambda$. In other words, all lattice vectors in Λ lie on hyperplanes of the form $\langle \cdot, \boldsymbol{y} \rangle \in \mathbb{Z}$.



Let $\boldsymbol{x}^* \in \Lambda/\{\boldsymbol{0}\}$ be the unknown shortest vector. We distinguish two case:

- *Case 1:* One has $\langle \boldsymbol{x}^*, \boldsymbol{y} \rangle \in \mathbb{Z}/\{\boldsymbol{0}\}$. The distance of the hyperplanes $\langle \cdot, \boldsymbol{y} \rangle = \mathbb{Z}$ to each other is $\frac{1}{\|\boldsymbol{y}\|_2}$, hence we have a lower bound of $\|\boldsymbol{x}^*\|_2 \geq \frac{1}{\|\boldsymbol{y}\|_2} \geq \frac{1}{f(n)}$. Our found lattice vector has length $\|\boldsymbol{x}\|_2 \leq f(n) \leq f(n)^2 \cdot \|\boldsymbol{x}^*\|_2$, hence $\boldsymbol{x}$ is the desired $f(n)^2$-approximation.
- *Case 2:* One has $\langle \boldsymbol{x}^*, \boldsymbol{y} \rangle = 0$. In this case, the shortest vector $\boldsymbol{x}^*$ lies in the sublattice $\Lambda' := \{\boldsymbol{x}' \in \Lambda : \langle \boldsymbol{x}', \boldsymbol{y} \rangle = 0\}$ of rank $n-1$. We apply our approximation algorithm for SVP recursively to that sublattice and inductively obtain a lattice vector $\boldsymbol{x}' \in \Lambda'/\{\boldsymbol{0}\}$ of length $\|\boldsymbol{x}'\|_2 \leq f(n-1)^2 \cdot \mathrm{SVP}(\Lambda') \leq f(n)^2 \cdot \mathrm{SVP}(\Lambda)$. Here we have used that f is non-decreasing. Implicitly, we also used Lemma 1.22 to argue that one can induce on a sublattice without any problem.

While we do not know in our algorithm in which case we are in, we can compare the length $\boldsymbol{x}$ with the length of the vector provided by Case 2 and return the shorter one. □

We should remark that the rank of the lattice $\{\boldsymbol{x} \in \Lambda \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0\}$ is indeed $n-1$, that means there are indeed $n-1$ linearly independent lattice vectors in the hyperplane $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$. The only prerequisite that is needed for this conclusion is that $\boldsymbol{y}$ is rational. But it is possible that the sublattice is a lot *sparser*, that means the determinant of the sublattice might be a lot larger than $\det(\Lambda)$.

1.7 The Hermite Normal Form

The question that we want to answer here is, how one can compute a lattice basis for the intersection of a lattice Λ with a subspace. For this section, we follow Chapter 4 and 5 in [Sch99]. Let us keep the situation a bit more general and consider a matrix $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) \in \mathbb{Q}^{n \times m}$ with $m \geq n$ and *full row rank*, that means $\mathrm{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) = \mathbb{R}^n$. We also want to consider the lattice $\Lambda(\boldsymbol{B}) = \{\sum_{i=1}^m \lambda_i \boldsymbol{b}_i \mid \lambda_1, \ldots, \lambda_m \in \mathbb{Z}\}$. Note that now the vectors spanning the lattice are not necessarily linearly independent. However, we will see that there is a lattice basis $\tilde{\boldsymbol{B}} \in \mathbb{Q}^{n \times n}$ so that $\Lambda(\tilde{\boldsymbol{B}}) = \Lambda(\boldsymbol{B})$.

The first question is, how can we change the matrix $\boldsymbol{B}$ without changing the generated lattice?

Definition 4. The following operations on a matrix $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) \in \mathbb{R}^{n \times m}$ are called *unimodular column operations*:

a) Exchanging columns $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ for $i \neq j$.

b) Replacing $\boldsymbol{b}_i$ by $-\boldsymbol{b}_i$.

c) Replacing $\boldsymbol{b}_i$ by $\boldsymbol{b}_i + \alpha \cdot \boldsymbol{b}_j$ for $j \neq i$ and some $\alpha \in \mathbb{Z}$.

The following is easy to see:

Corollary 1.24. *Let $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ be a regular matrix and let $\tilde{\boldsymbol{B}}$ be the matrix after any number of unimodular column operations. Then $\Lambda(\boldsymbol{B}) = \Lambda(\tilde{\boldsymbol{B}})$.*

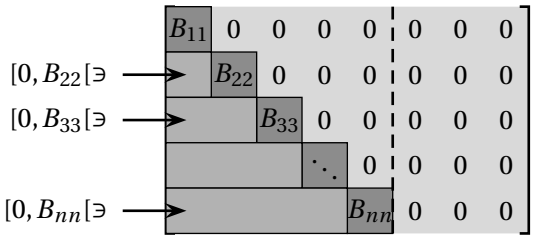
The question arises whether there is a structurally rich "normal form" that one can bring every lattice basis into. And indeed, this normal form exists:

Definition 5. Let $\boldsymbol{B} \in \mathbb{Q}^{n\times m}$ be a matrix. Then we say that $\boldsymbol{B}$ is in *Hermite normal form* if

i) One has $\boldsymbol{B} = (\boldsymbol{L}, \boldsymbol{0})$ where $\boldsymbol{L}$ is a lower triangular matrix

ii) $B_{ij} \geq 0$ for all $i, j \in [n]$

iii) Each diagonal entry B_{ii} is the unique maximum entry for that row i



The first observation is that as $\boldsymbol{B}$ is rational, we can scale $\boldsymbol{B}$ so that $\boldsymbol{B} \in \mathbb{Z}^{n\times m}$. Before we move on with the general case, let us discuss the special case that $\boldsymbol{B} \in \mathbb{Z}^{1\times m}$ as only *one row*. For this case, we know that the *Euclidean algorithm* can add and subtract entries until only one non-zero entry is left which can be moved to the leftmost entry. Note that the non-zero entry will be precisely the greatest common divisor:

$$(B_{11}, \ldots, B_{1m}) \overset{\text{unimodular column operations}}{\longmapsto} (\gcd, 0, \ldots, 0)$$

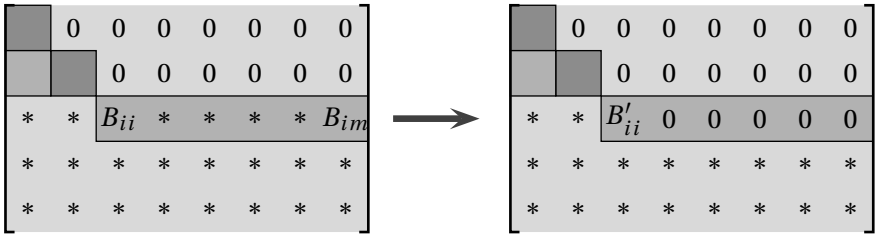
Note that the emerging row is indeed in Hermite normal form. We will now see an algorithm that is a generalization of the Euclidean algorithm to the matrix world:

Theorem 1.25. *There is a algorithm that takes any matrix $\boldsymbol{B} \in \mathbb{Z}^{n\times m}$ as input and performs poly$(n, m, \log \|\boldsymbol{B}\|_\infty)$ many unimodular row operations to obtain $\tilde{\boldsymbol{B}}$ in Hermite normal form.*

Proof. Let $\boldsymbol{B} \in \mathbb{Z}^{n\times m}$ be the input matrix. We will now apply unimodular column operations until $\boldsymbol{B}$ is in Hermite normal form:

(1) FOR $i = 1$ TO n DO

(2) Perform the Euclidean algorithm to entries $(B_{ii}, B_{i,i+1}, \ldots, B_{i,m})$ (actually to columns $i, \ldots, m$) and obtain entries $(B'_{ii}, 0, \ldots, 0)$



(3) Add multiples of column i to columns $1,\dots,i-1$ until $0 \le B_{ij} < B_{ii}$ for all $j = 1,\dots,i-1$

We know that Euclid's algorithm takes polynomial time, hence each iteration of (2) takes a polynomial number of column operations. It is somewhat clear that the matrix that we get at the end will indeed be in Hermite normal form. □

The above argument shows that the number of iterations is bounded. But it does not immediately imply that the encoding length of any intermediate number is polynomial as well. In fact, it takes quite some effort. To get that result one has to do computations modulo M where M is the largest absolute value of any subdeterminant of $\boldsymbol{B}$. The details can be found in [Sch99].

Corollary 1.26. *For any integral matrix $\boldsymbol{A} \in \mathbb{Z}^{n\times m}$ one can compute a unimodular matrix $\boldsymbol{U} \in \mathbb{Z}^{n\times n}$ in time poly$(n, m, \log \|\boldsymbol{A}\|_\infty)$ so that $\boldsymbol{AU} = (\boldsymbol{B}, \boldsymbol{0})$ is in Hermite normal form.*

One can also show the following:

Theorem 1.27. *The Hermite normal form of every matrix is unique.*

Again, for a proof, see [Sch99].

Lemma 1.28. *Let $\Lambda(\boldsymbol{B})$ be a lattice with $\boldsymbol{B} \in \mathbb{Z}^{n\times n}$ and let $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} = \boldsymbol{0}\}$ with $\boldsymbol{A} \in \mathbb{Q}^{m\times n}$ be a subspace. Then there is a polynomial time algorithm to compute a lattice basis $\boldsymbol{B}'$ so that $\Lambda(\boldsymbol{B}') = \{\boldsymbol{x} \in \Lambda \mid \boldsymbol{Ax} = \boldsymbol{0}\}$.*

Proof. Suppose that $k := \dim(\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} = \boldsymbol{0}\})$ is the dimension of the subspace. Rotate the lattice so that the subspace is spanned by the last k standard basis vectors, that means so that $\{\boldsymbol{x} \mid \boldsymbol{Ax} = \boldsymbol{b}\} = \operatorname{span}(\boldsymbol{e}_{n-k+1},\dots,\boldsymbol{e}_n)$. Then compute the Hermite normal form of $\boldsymbol{B}$. The vectors $\boldsymbol{b}_{n-k+1},\dots,\boldsymbol{b}_n$ will span the sublattice. □

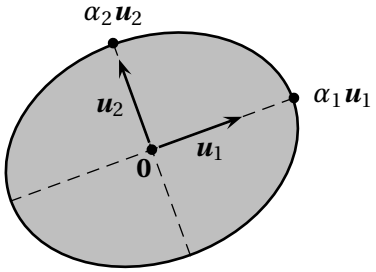
Note that obviously $\Lambda(\boldsymbol{B}')$ would not have full rank in this case.

1.8 Minkowski's 2nd Theorem

Now we want to give the proof of *Minkowski's Second Theorem* that we had postponed so far. In the presentation, we follow [Reg09]. Take any *orthonormal basis* $\boldsymbol{u}_1,\dots,\boldsymbol{u}_n \in \mathbb{R}^n$ and any positive coefficients $\alpha_1,\dots,\alpha_n > 0$. Then

$$\mathcal{E} = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \sum_{i=1}^{n} \frac{1}{\alpha_i^2} \cdot \langle \boldsymbol{x}, \boldsymbol{u}_i\rangle^2 \le 1\right\}$$

is an *ellipsoid*. We will discuss more details in the next Chapter, but for now we will just use that ellipsoids are convex symmetric bodies and their volume is $\operatorname{vol}(\mathcal{E}) = \operatorname{vol}(\mathcal{B}(\boldsymbol{0}, 1)) \cdot \prod_{i=1}^{n} \alpha_i$.



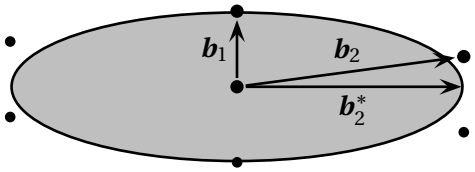
The intuition is that Minkowski's First Theorem gives an upper bound of $\sqrt{n} \cdot \det(\Lambda)^{1/n}$ on the length of the shortest vector; Minkowski's Second Theorem gives the stronger statement that this bound is even achieved for the *geometric average* of the successive shortest vectors.

Theorem 1.29 (Minkowski's Second Theorem)**.** *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$ one has*

$$\Big(\prod_{i=1}^{n} \lambda_i(\Lambda)\Big)^{1/n} \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}.$$

Proof. We abbreviate $\lambda_i := \lambda_i(\Lambda)$. Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \Lambda \setminus \{\mathbf{0}\}$ be the vectors that attain the successive minima, that means $\lambda_i = \|\boldsymbol{b}_i\|_2$ with $\lambda_1 \leq \ldots \leq \lambda_n$. Let $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ be the Gram Schmidt orthogonalization (in that order). We consider the ellipsoid

$$\mathcal{E} = \Big\{\boldsymbol{x} \in \mathbb{R}^n \mid \Big(\frac{\langle \boldsymbol{x}, \boldsymbol{b}_i^* \rangle}{\|\boldsymbol{b}_i^*\|_2 \cdot \lambda_i}\Big)^2 \leq 1\Big\}$$



Let $\operatorname{int}(\mathcal{E}) = \{\boldsymbol{x} \mid \ldots < 1\}$ be the interior of that elipsoid. We claim that $\operatorname{int}(\mathcal{E}) \cap \Lambda = \{\mathbf{0}\}$. Take any lattice vector $\boldsymbol{x} \in \Lambda \setminus \{\mathbf{0}\}$. Let k be maximal so that $\lambda_k \leq \|\boldsymbol{x}\|_2 < \lambda_{k+1}$. Note that this means that $\boldsymbol{x} \in \operatorname{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k) = \operatorname{span}(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_k^*)$ since otherwise, we could have chosen $\boldsymbol{x}$ instead of $\boldsymbol{b}_{k+1}$ and the value of λ_{k+1} would have been shorter than it is. Now we can bound

$$\sum_{i=1}^{n} \Big(\frac{\langle \boldsymbol{x}, \boldsymbol{b}_i^* \rangle}{\|\boldsymbol{b}_i\|_2 \cdot \lambda_i}\Big)^2 \overset{\substack{\langle \boldsymbol{x}, \boldsymbol{b}_i^* \rangle = 0 \ \forall i > k \\ \lambda_1 \leq \ldots \leq \lambda_k}}{\geq} \frac{1}{\lambda_k^2} \underbrace{\sum_{i=1}^{k} \langle \boldsymbol{x}, \frac{\boldsymbol{b}_i^*}{\|\boldsymbol{b}_i^*\|_2} \rangle^2}_{= \|\boldsymbol{x}\|_2^2} = \frac{\|\boldsymbol{x}\|_2^2}{\lambda_k} \geq 1$$

This implies that indeed $\boldsymbol{x} \notin \operatorname{int}(\mathcal{E})$. Since $\mathcal{E}$ does not have a lattice point in its interior, *Minkowski's First Theorem* gives an upper bound on its volume:

$$2^n \cdot \det(\Lambda) \geq \operatorname{vol}(\mathcal{E}) = \operatorname{vol}(\mathcal{B}(\mathbf{0}, 1)) \cdot \prod_{i=1}^{n} \lambda_i \geq \Big(\frac{2}{\sqrt{n}}\Big)^n \cdot \prod_{i=1}^{n} \lambda_i$$

In $(*)$ we have used that the box $[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]^n$ is contained in $\mathcal{B}(\mathbf{0}, 1)$ and has volume $(\frac{2}{\sqrt{n}})^n$. Rearranging then gives the claim. □

1.9 Exercises

Exercise 1.1. Let $\Lambda = \Lambda(\boldsymbol{B})$ with $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ be a lattice. Show that for any $\varepsilon > 0$ there is a radius $R := R(\varepsilon, n, \boldsymbol{B})$ so that

$$(1 - \varepsilon) \cdot \frac{\operatorname{vol}(\mathcal{B}(\mathbf{0}, R))}{\det(\Lambda)} \leq |\mathcal{B}(\mathbf{0}, R) \cap \Lambda| \leq (1 + \varepsilon) \cdot \frac{\operatorname{vol}(\mathcal{B}(\mathbf{0}, R))}{\det(\Lambda)}$$

Exercise 1.2. Let $K \subseteq \mathbb{R}^n$ be a symmetric convex set with $\operatorname{vol}(K) > k \cdot 2^n$.

a) Show that $|K \cap \mathbb{Z}^n| \geq k$.

b) Can you strengthen the claim to argue that you get k linearly independent points?

Exercise 1.3. Show that there is a lattice $\Lambda = \Lambda(\boldsymbol{B}) \subseteq \mathbb{R}^n$ with $|\det(\boldsymbol{B})| = 1$ and $\text{SVP}(\Lambda) \geq \Omega(\sqrt{n})$.
Remark: This is a hard exercise that gives extra points. I currently do not have a solution (though there should be a proof somewhere in the literature). I would suspect that one could take each entry B_{ij} independently at random from a Gaussian distribution $N(0,1)$ and then scale the lattice accordingly. I would very much appreciate a slick proof for this.

Exercise 1.4. This is an application of Dirichlet's Theorem: Let $\boldsymbol{a} \in]0,1]^n$ be a real vector and consider the hyperplane $H := \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{a}, \boldsymbol{x} \rangle = 0\}$. Then there is a rational vector $\tilde{\boldsymbol{a}} \in \frac{\mathbb{Z}^n}{q}$ with $q \leq (2nR)^n$ so that $\tilde{H} := \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \tilde{\boldsymbol{a}}, \boldsymbol{x} \rangle = 0\}$ satisfies the following:

$$\forall \boldsymbol{x} \in \{-R, \ldots, R\}^n : \boldsymbol{x} \in H \Rightarrow \boldsymbol{x} \in \tilde{H}.$$

Exercise 1.5. For a lattice $\Lambda \subseteq \mathbb{R}^n$, let $\lambda_i(\Lambda) := \min\{r \geq 0 \mid \dim(\text{span}(\mathcal{B}(\boldsymbol{0}, r) \cap \Lambda)) \geq i\}$ the minimum radius of a ball that contains at least i linearly independent lattice vectors. Show that there exists a vector $\boldsymbol{t} \in \mathbb{R}^n$ with $\text{dist}(\boldsymbol{t}, \Lambda) \geq \frac{1}{2}\lambda_n(\Lambda)$. Here $\text{dist}(\boldsymbol{t}, \Lambda) := \min\{\|\boldsymbol{x} - \boldsymbol{t}\|_2 \mid \boldsymbol{x} \in \Lambda\}$ gives the Euclidean distance of $\boldsymbol{t}$ to the lattice.

Exercise 1.6. Let $S \subseteq \mathbb{R}^n$ be a measurable, compact set. For $\text{vol}(S) > k$ for some $k \in \mathbb{Z}_{\geq 0}$. Then there are points $\boldsymbol{s}_0, \ldots, \boldsymbol{s}_k \in S$ with $\boldsymbol{s}_i - \boldsymbol{s}_j \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$ for all $i \neq j$.

Exercise 1.7. Let $\Lambda \subseteq \mathbb{R}^n$ be a full rank lattice. Show that $\lambda_1(\Lambda) \cdot \lambda_n(\Lambda^*) \geq 1$, where Λ^* is the dual lattice.

Exercise 1.8. Let $\Lambda := \Lambda(\boldsymbol{B}) \subseteq \mathbb{R}^n$ be a full rank lattice for the basis $\boldsymbol{B}$. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \Lambda \setminus \{\boldsymbol{0}\}$ be linear independent vectors. Then $\max\{\|\boldsymbol{v}_1\|_2, \ldots, \|\boldsymbol{v}_n\|_2\} \geq \|\boldsymbol{b}_n^*\|_2$ where $\boldsymbol{b}_n^*$ is the "last" vector in the Gram-Schmidt orthogonalization of $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$.
Hint: Read again the proof that $\min\{\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*\}$ gives a lower bound on $\text{SVP}(\Lambda)$.

Exercise 1.9. Prove that for any lattice $\Lambda \subseteq \mathbb{R}^n$, one has $\lambda_1(\Lambda) \cdot \lambda_1(\Lambda^*) \leq n$.
Remark: A stronger theorem of Banaszczyk [Ban93a] shows that even $\lambda_1(\Lambda) \cdot \lambda_n(\Lambda^*) \leq n$. This has an important consequence. Consider the following computational problem: Given a lattice Λ and a parameter K, distinguish the cases $\text{SVP}(\Lambda) \leq K$ and $\text{SVP}(\Lambda) \geq n \cdot K$. The consequence of this exercise is that this problem is in $\mathbf{NP} \cap \mathbf{coNP}$ in the sense that one can give an efficiently checkable proof for $\text{SVP}(\Lambda) \leq K$ (simply give me a short vector) and one can also certify is $\text{SVP}(\Lambda) \geq n \cdot K$ (give me the short dual basis). The remarkable think is that this gap problem is not known to be in $\boldsymbol{P}$.

Exercise 1.10. Consider the matrix

$$\boldsymbol{B} = \begin{pmatrix} 2 & 3 & 4 \\ 2 & 4 & 6 \end{pmatrix}$$

Compute the Hermite Normal form of $\boldsymbol{B}$.

Exercise 1.11. Let $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n$ be a vector of integer numbers. The original *Euclidean algorithm* does the following:

(1) REPEAT

(2) Select the index i with $|a_i|$ minimal.

(3) For all $j \neq i$ replace a_j by $\min\{|a_j + z \cdot a_i| \mid z \in \mathbb{Z}\}$.

Prove that the algorithm terminates after at most $O(\log \|\boldsymbol{a}\|_\infty)$ many iterations.

Exercise 1.12. Let $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{Z}^m$ with $m \leq n$ where $\boldsymbol{A}$ has full row rank. Show that in polynomial time one can compute a vector $\boldsymbol{x} \in \mathbb{Z}^n$ with $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ (or decide that no such vector exists).
Remark: Use Cor. 1.26.

Exercise 1.13. Let $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ be any regular matrix and let $\Lambda := \Lambda(\boldsymbol{B})$ be the spanned lattice. Prove that there is a regular matrix $\tilde{\boldsymbol{B}}$ so that $\tilde{\Lambda} := \Lambda(\tilde{\boldsymbol{B}})$ is a sublattice of Λ (that means $\tilde{\Lambda}$ is a lattice and $\tilde{\Lambda} \subseteq \Lambda$) and the *orthogonality defect* satisfies $\gamma(\tilde{\boldsymbol{B}}) \leq n^{O(n)}$.
Hint: Use Minkowski's Second Theorem.
Remark: There is also an actual basis — the Khorkine Zolotarav basis — that has orthogonality defect at most $n^{O(n)}$. But that his harder to prove.

Exercise 1.14. Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice. Assume that $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \Lambda$ are linearly-independent and minimize $|\det(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)|$. Prove that $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ are indeed a *basis* of Λ.

Chapter 2

Integer Programming in Fixed Dimension

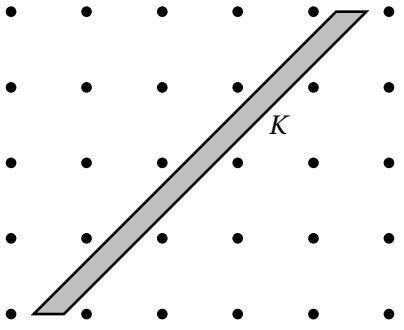
Integer programming is one of the most powerful and most useful problems in discrete optimization.

INTEGER PROGRAMMING (IP)
Input: A linear inequality system $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$
Goal: Find a point $\boldsymbol{x} \in K \cap \mathbb{Z}^n$ where $K := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$

This problem is among the first problems that were shown to be **NP**-hard. The fundamental practical importance comes from the fact that the standard approach for operations research practicioners is to model whatever problem appears in their real-world application as an integer linear program and then solve it using quite sophisticated software tools. In this chapter, we will look at the problem from a purely theoretical perspective. For a more detailed treatment, we refer to the survey of Kannan [Kan87b].

As the integer programming problem is **NP**-hard, there is no hope for a polynomial time algorithm, but it is natural to ask for whether the problem can be solved in time $f(n) \cdot \text{poly}(n, \log \|\boldsymbol{A}\|_\infty, \log \|\boldsymbol{b}\|_\infty)$ where $f(n)$ will be some exponentially growing function of the dimension; here we have implicitly assumed that $\boldsymbol{A}$ and $\boldsymbol{b}$ are scaled to be integers. In other words: *can we solve integer programming in polynomial time when the dimension is some fixed constant?* The affirmative answer due to Lenstra [Len83] is highly non-trivial and gives some beautiful convex geometric insights. To get rid of technical arguments, we will assume that K is bounded — one could achieve this easily by intersecting K with a bounding box $[-M, M]^n$ where M can be chosen so large that $K \cap \mathbb{Z}^n \neq \emptyset$ if and only $K \cap \{-M, \ldots, M\}^n \neq \emptyset$.

The idea behind Lenstra's algorithm is to solve the problem recursively where we split an n-dimensional problem into $g(n)$ many $(n-1)$-dimensional subproblems where g should be a function depending only on the dimension. The first naive approach would be to select a coordinate i and recurse on all subproblems of the form $K \cap \{\boldsymbol{x} \in \mathbb{R}^n \mid x_i = \delta\}$ for $\delta \in \mathbb{Z}$. But it is not hard to come up with a polytope K with $K \cap \mathbb{Z}^n = \emptyset$ that is very thin while it is long in each coordinate direction.



So this approach does not immediately work. On the other hand, one gets the impression that if K does not contain an integer point it would be thin in *some direction*. We want to make that more formal:

For a vector $\boldsymbol{c} \in \mathbb{Z}^n$, let

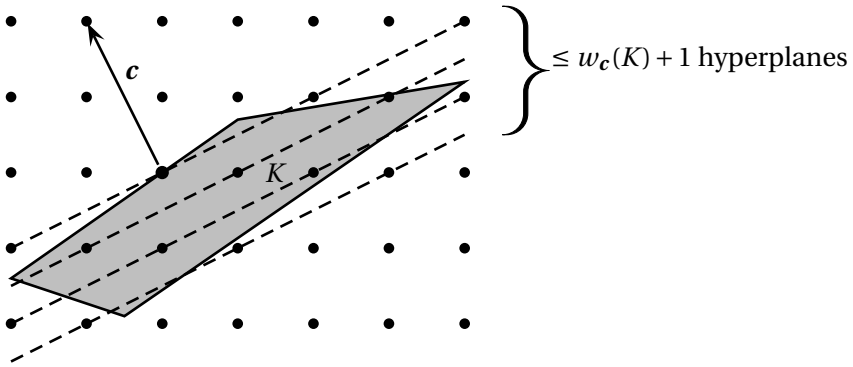
$$w_{\boldsymbol{c}}(K) := \max\{\boldsymbol{c}\boldsymbol{x} - \boldsymbol{c}\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{y} \in K\}$$

be the *width in direction* $\boldsymbol{c}$. Moreover, let

$$w(K) := \min_{\boldsymbol{c} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}} w_{\boldsymbol{c}}(K)$$

be the *integer width* of K. Now, we observe that this thin direction $\boldsymbol{c}$ is exactly what we need to recurse on $(n-1)$-dimensional subproblems:

Observation 2.1. Given K and $\boldsymbol{c} \in \mathbb{Z}^n$. All points in $K \cap \mathbb{Z}^n$ are contained in at most $w_{\boldsymbol{c}}(K) + 1$ many hyperplanes of the form $K \cap \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{c}\boldsymbol{x} = \delta\}$ with $\delta \in \mathbb{Z}$.



It is important to note that $w(K)$ is not the geometric width — it is the geometric width *times* the length $\|\boldsymbol{c}\|_2$. In order to show that $w(K)$ is small one has to find a *short* vector $\boldsymbol{c} \in \mathbb{Z}^n$ so that K is thin in direction $\boldsymbol{c}$. The fundamental theorem behind Lenstra's algorithm is a theorem that tells us that each polytope without an integral point must have a direction with bounded integer width:

Theorem 2.2 (Khinchine's Flatness Theorem)**.** *Let $K = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ be a polytope. Then in polynomial time one can find:*

(A) Either a point $\boldsymbol{x}^* \in K \cap \mathbb{Z}^n$
(B) Or a direction $\boldsymbol{c} \in \mathbb{Z}^n$ with $w_{\boldsymbol{c}}(K) \leq f(n)$ where $f(n) \leq 2^{O(n^2)}$.

In the claim, with "polynomial time", we mean that the running time is bounded by $\text{poly}(n, \log \|\boldsymbol{A}\|_\infty, \log \|\boldsymbol{b}\|_\infty)$ given that the inequality system has been multiplied by the least common denominator so that $\boldsymbol{A}$ and $\boldsymbol{b}$ have only integral entries. We should remark that the restriction to polytopes is not really needed for a non-constructive bound on $f(n)$. Note that there are stronger bounds for the integer width known. For any convex body, $f(n) \leq O(n^{4/3} \cdot \log^{O(1)}(n))$. If K is indeed a polytope with n^c facets, then $f(n) \leq O(cn \log n)$ is enough. On the other hand, there are constructions showing that $f(n) \geq \Omega(n)$ is needed. The article by Dadush [Dad14] gives a good overview.

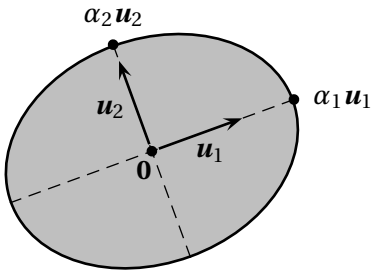
2.1 John's Theorem

One of the most important results in convex geometry is *John's Theorem*, which says that any convex set "looks" like an *ellipsoid*, up to a factor of n. For the sake of simplicity, we give the definition only for

ellipsoids with center $\mathbf{0}$ — a general ellipsoid is simply obtained by translation. Take any *orthonormal basis* $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n \in \mathbb{R}^n$ and any positive coefficients $\alpha_1, \ldots, \alpha_n > 0$. Then

$$\mathcal{E} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \sum_{i=1}^{n} \frac{1}{\alpha_i^2} \cdot \langle \boldsymbol{x}, \boldsymbol{u}_i \rangle^2 \leq 1 \right\}$$

is an ellipsoid.



The basis vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ are also called the *axis* of the ellipsoid. In particular, the points $\alpha_1 \boldsymbol{u}_1, \ldots, \alpha_n \boldsymbol{u}_n$ lie on the boundary of the ellipsoid. If $\alpha_1 = \ldots = \alpha_n = 1$, then $\mathcal{E} = \mathcal{B}(\mathbf{0}, 1)$. Recall that $\mathcal{B}(\boldsymbol{c}, r) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{c}\|_2 \leq r\}$ is the ball of radius r with center $\boldsymbol{c} \in \mathbb{R}^n$. Note that we can also write the ellipsoid as

$$\mathcal{E} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{x}\boldsymbol{x}^T, \underbrace{\sum_{i=1}^{n} \frac{1}{\alpha_i^2} \boldsymbol{u}_i \boldsymbol{u}_i^T}_{=:\boldsymbol{A}} \rangle \leq 1 \right\} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \leq 1 \right\}$$

The Eigenvalues of the matrix $\boldsymbol{A}$ are exactly $\frac{1}{\alpha_1}, \ldots, \frac{1}{\alpha_n} > 0$, hence A is positive definite and symmetric. Note that both representations are equivalent since every symmetric positive definite matrix $\boldsymbol{A}$ can be written as $\boldsymbol{A} = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T$ where $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ gives the orthonormal basis of Eigenvectors. In fact, there is a third possible definition: an ellipsoid is the image of the unit ball under a linear map. To be more precise, we can use the *linear map* $T : \mathbb{R}^n \to \mathbb{R}^n$ with $T(\boldsymbol{y}) := \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{u}_i$, so that $T(\boldsymbol{e}_i) = \alpha_i \cdot \boldsymbol{u}_i$ and hence $\mathcal{E} = \{T(\boldsymbol{y}) : \|\boldsymbol{y}\|_2 \leq 1\}$.

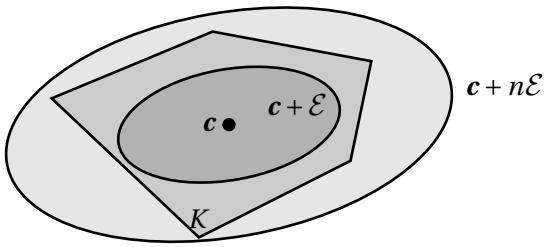
Next, note that the *volume* of an ellipsoid is

$$\mathrm{vol}(\mathcal{E}) = \mathrm{vol}(\mathcal{B}(\mathbf{0}, 1)) \cdot \prod_{i=1}^{n} \alpha_i.$$

This is also easy to see: just stretch the unit $\mathcal{B}(\mathbf{0}, 1)$ along $\boldsymbol{u}_i$ by a factor of α_i. Iterating this argument or all axis gives exactly the claimed volume.

Theorem 2.3 (John)**.** *For any convex body $K \subseteq \mathbb{R}^n$ there is a translate of an ellipsoid so that*

$$\boldsymbol{c} + \mathcal{E} \subseteq K \subseteq \boldsymbol{c} + n \cdot \mathcal{E}.$$



We want to give at least a proof of this theorem, at least with a factor Cn for some constant $C > 0$ instead of the tight bound.

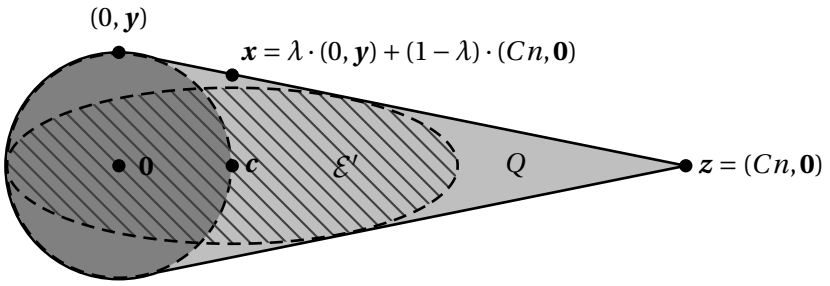
Proof. Let $\mathcal{E}$ be the ellipsoid with *maximum volume* that is contained in K (as usually, this maximum exists by a compactness argument[1]). After a linear transformation which does not change the validity of the claim, we can assume that that this maximum volume ellipsoid is indeed the unit ball, that means $\mathcal{E} = \mathcal{B}(\mathbf{0}, 1)$. Assuming that K is not contained in $Cn \cdot \mathcal{B}(\mathbf{0}, 1)$ we do know that there has to be a point $\boldsymbol{z} \in K$ with $\|\boldsymbol{z}\|_2 = Cn$. After rotating K, we can assume that $\boldsymbol{z} = Cn \cdot \boldsymbol{e}_1$, We define a convex body

$$Q := \operatorname{conv}(\mathcal{B}(\mathbf{0}, 1) \cup \{Cn \cdot \boldsymbol{e}_1\})$$

for which we surely know that it is contained in K. We want to find an ellipsoid $\mathcal{E}'$ that is contained in $Q \subseteq K$ and that has a larger volume than $\mathcal{B}(\mathbf{0}, 1)$. The idea is to stretch the unit ball along the x_1-axis and to sqeeze it along all the other axis. Moreover, we choose a new center $\boldsymbol{c} := \boldsymbol{e}_1$ that is moved slightly towards the point $Cn \cdot \boldsymbol{e}_1$. Formally, we choose coefficients $\alpha_1 := 2$ as well as $\alpha_2 := \ldots = \alpha_n := \sqrt{\frac{n}{n+\frac{1}{10}}} = 1 - \Theta(\frac{1}{n})$. Our enlarged ellipsoid is then

$$\mathcal{E}' := \left\{\boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^{n} \frac{1}{\alpha_i^2}(x_i - c_i)^2 \leq 1\right\} = \left\{\boldsymbol{x} \in \mathbb{R}^n : \underbrace{\frac{1}{2^2}(x_1 - 1)^2 + (1 + \frac{1}{10n})\sum_{i=2}^{n} x_i^2}_{=:f(\boldsymbol{x})} \leq 1\right\}$$

using that $\frac{1}{\alpha_i^2} = 1 + \frac{1}{10n}$ for $i \geq 2$.



It is easy to check that the volume of ellipsoid $\mathcal{E}'$ satisfies

$$\frac{\operatorname{vol}(\mathcal{E}')}{\operatorname{vol}(\mathcal{B}(\mathbf{0}, 1))} = \prod_{i=1}^{n} \alpha_i = 2 \cdot \underbrace{\left(1 - \frac{1}{10n + 100}\right)^{(n-1)/2}}_{\approx \exp(-1/5)} > 1$$

To prove that indeed $\mathcal{E}' \subseteq Q \subseteq K$ it suffices to prove that the boundary of Q is not in the ellipsoid.
Claim: Let $\boldsymbol{x} = \lambda \cdot (0, \boldsymbol{y}) + (1 - \lambda) \cdot (Cn, \mathbf{0})$ with $0 \leq \lambda \leq 1$ and $\boldsymbol{y} \in \mathbb{R}^{n-1}$, $\|\boldsymbol{y}\|_2 = 1$. Then $f(\boldsymbol{x}) > 1$.
Proof: We estimate that

$$f(\boldsymbol{x}) = \frac{1}{4}((1 - \lambda) \cdot Cn - 1)^2 + (1 + \frac{1}{10n}) \underbrace{\sum_{i=1}^{n-1} (\lambda y_i)^2}_{=\lambda^2} = \frac{1}{4} \cdot ((1 - \lambda)Cn - 1)^2 + (1 + \frac{1}{10n})\lambda^2$$

[1] Moreover we show that the volume increases in each step by a constant factor – so we don't actually rely on the compactness argument.

If $\lambda \geq 1 - \frac{1}{30n}$, then already the 2nd summand is at least $(1 + \frac{1}{10n}) \cdot (1 - \frac{1}{30n})^2 > 1$. If $\lambda \leq 1 - \frac{1}{30n}$, then the first summand is

$$\frac{1}{4} \cdot (\frac{1}{30n} \cdot Cn - 1)^2 > 1$$

for C large enough. □

The factor n is best possible in general and is attained for example for the n-dimensional simplex. On the other hand, if K is symmetric, then the factor can be improved to $\sqrt{n}$. Here, we call a body K *symmetric* if $\boldsymbol{x} \in K \Leftrightarrow -\boldsymbol{x} \in K$. Note that in particular, this enforces that $\mathbf{0}$ is the center of the ellipsoids.

Theorem 2.4 (John)**.** *For any symmetric convex body $K \subseteq \mathbb{R}^n$ there is an ellipsoid $\mathcal{E}$ so that $\mathcal{E} \subseteq K \subseteq \sqrt{n} \cdot \mathcal{E}$.*

We will leave the proof for the exercises.

2.2 A flatness theorem for Ellipsoids

Now we know that any convex body can be approximated up to a factor n with an ellipsoid. In particular this implies that it suffices to prove the flatness theorem for ellipsoids — then it will follow with a factor-n loss also for the general convex body. Instead of directly searching for an integer point in an ellipsoid $\mathcal{E}$ we will apply a linear transformation to it so that it becomes a unit ball. This also transforms the integer lattice into a general lattice. It remains to prove the following:

Theorem 2.5. *Let $\Lambda = \Lambda(\boldsymbol{B})$ be any lattice and let $K = \mathcal{B}(\boldsymbol{a}, 1)$ be a unit ball with center $\boldsymbol{a}$. Then in polynomial time one can find*

(A) either a point $\boldsymbol{x}^* \in K \cap \Lambda$
(B) or a direction $\boldsymbol{c} \in \mathbb{R}^n$ so that the points in $K \cap \Lambda$ are covered by at most $2^{O(n^2)}$ hyperplanes of the form $\boldsymbol{cx} = \delta$ with $\delta \in \mathbb{Z}$ (actually we will have $\boldsymbol{c} \in \Lambda^*$ and $\|\boldsymbol{c}\|_2 \leq 2^{O(n^2)}$).

Proof. We can use the LLL-algorithm and replace the initial lattice basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ with an LLL-reduced basis. In particular we know that the *orthogonality defect* is

$$\gamma(\boldsymbol{B}) = \prod_{i=1}^{n} \frac{\|\boldsymbol{b}_i\|_2}{\|\boldsymbol{b}_i^*\|_2} \leq 2^{n^2/2} \tag{2.1}$$

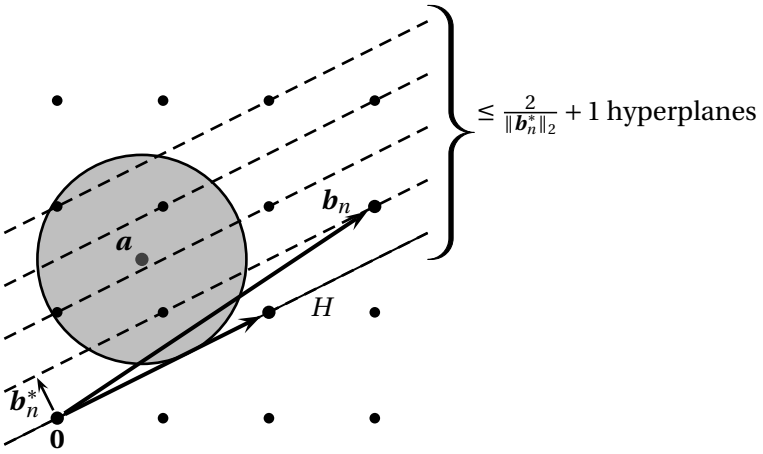
Recall that $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ is the Gram-Schmidt orthogonalization of $\boldsymbol{B}$ with respect to the order $1, \ldots, n$. We know that the quantity $\prod_{i=1}^{n} \|\boldsymbol{b}_i^*\|_2 = \det(\Lambda)$ is independent of the basis and in particular of the *order* of the basis vectors. Now, let us reorder the basis vectors so that $\|\boldsymbol{b}_1\|_2 \leq \ldots \leq \|\boldsymbol{b}_n\|_2$. Note that this basis is not necessarily LLL-reduced anymore. Yet, we know that it's orthogonality defect has not changed and (2.1) still holds. Now we can make an easy choice for the hyperplane direction:
Claim. If $\|\boldsymbol{b}_n\|_2 \leq \frac{1}{n}$, then we can find a point $\boldsymbol{x}^* \in K \cap \Lambda$ in polynomial time.
Proof. Let us write the center of the ball as $\boldsymbol{a} = \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i$ with $\lambda_i \in \mathbb{R}$. Then our choice is simply $\boldsymbol{x}^* := \sum_{i=1}^{n} \lfloor \lambda_i \rfloor \cdot \boldsymbol{b}_i \in \Lambda$. The distance of that lattice point to the center of the ball is $\|\boldsymbol{x}^* - \boldsymbol{a}\|_2 \leq \sum_{i=1}^{n} \|\boldsymbol{b}_i\|_2 \leq n \cdot \|\boldsymbol{b}_n\|_2 \leq n \cdot \frac{1}{n} = 1$ using that $\boldsymbol{b}_n$ was the longest basis vector. Then $\boldsymbol{x}^* \in \mathcal{B}(\boldsymbol{a}, 1)$. □
Claim. If $\|\boldsymbol{b}_n\|_2 \geq \frac{1}{n}$, then

$$K \cap \Lambda \subseteq \bigcup_{\delta=\delta_{\min}}^{\delta_{\max}} \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{b}_n^*, \boldsymbol{x} \rangle = \delta\}$$

with $\delta_{\max} - \delta_{\min} \le 2n \cdot 2^{n^2/2}$.
Proof. Consider the $(n-1)$-dimensional subspace

$$H = \operatorname{span}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n-1}\} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{b}_n^*, \boldsymbol{x} \rangle = 0\}$$

We have $H + \boldsymbol{b}_n = H + \boldsymbol{b}_n^*$ hence $\Lambda \subseteq \bigcup_{\delta \in \mathbb{Z}} (H + \delta \cdot \boldsymbol{b}_n) = \bigcup_{\delta \in \mathbb{Z}} (H + \delta \cdot \boldsymbol{b}_n^*)$.



Since each of the factors defining the orthogonality defect is at least 1, we have $\frac{\|\boldsymbol{b}_n\|_2}{\|\boldsymbol{b}_n^*\|_2} \le \gamma(\boldsymbol{B}) \le 2^{n^2/2}$. To estimate the number of hyperplanes that intersect $\mathcal{B}(\boldsymbol{a}, 1)$, simply observe that the distance of consecutive hyperplanes is $\|\boldsymbol{b}_n^*\|_2 \ge 2^{-n^2/2} \|\boldsymbol{b}_n\|_2 \ge \frac{1}{n} \cdot 2^{n^2/2}$. On the other hand, the distance of the two outermost hyperplanes that still intersect $\mathcal{B}(\boldsymbol{a}, 1)$ is at most 2.

Now, let $\boldsymbol{c} := \frac{\boldsymbol{b}_n^*}{\|\boldsymbol{b}_n^*\|_2^2}$. Any lattice vector $\boldsymbol{x} \in \Lambda$ can be written in the form $\boldsymbol{x} = \lambda \cdot \boldsymbol{b}_n^* + \boldsymbol{h}$ with $\lambda \in \mathbb{Z}$ and $\boldsymbol{h} \in H$. Then

$$\langle \boldsymbol{c}, \boldsymbol{x} \rangle = \frac{1}{\|\boldsymbol{b}_n^*\|_2^2} \cdot \langle \boldsymbol{b}_n^*, \lambda \cdot \boldsymbol{b}_n^* + \boldsymbol{h} \rangle = \frac{1}{\|\boldsymbol{b}_n^*\|_2^2} \cdot \langle \lambda \cdot \boldsymbol{b}_n^*, \boldsymbol{b}_n^* \rangle = \lambda \in \mathbb{Z}.$$

Hence $\boldsymbol{c} \in \Lambda^*$ is in the dual lattice. Note that $\|\boldsymbol{c}\|_2 = \frac{1}{\|\boldsymbol{b}_n^*\|_2} \le 2^{O(n^2)}$. □

Remember that the *dual lattice* is the set of vectors $\boldsymbol{c} \in \mathbb{R}^n$ so that $\boldsymbol{c}\boldsymbol{x} \in \mathbb{Z}$ for all $\boldsymbol{x} \in \Lambda$. From a more abstract point of view, the last theorem is actually proving that either a reduced basis has only short vectors or there is a short vector in the dual lattice.

Corollary 2.6. *Let $K = \boldsymbol{a} + \mathcal{E}$ be an ellipsoid with center $\boldsymbol{a}$. Then in polynomial time one can find*

(A) either a point $\boldsymbol{x}^* \in K \cap \mathbb{Z}^n$
(B) or a direction $\boldsymbol{c} \in \mathbb{Z}^n$ so that the points in $K \cap \mathbb{Z}^n$ are covered by at most $2^{O(n^2)}$ hyperplanes of the form $\boldsymbol{c}\boldsymbol{x} = \delta$ with $\delta \in \mathbb{Z}$ (that means $w_{\boldsymbol{c}}(K) \le 2^{O(n^2)}$).

Proof. We can write the ellipsoid in the form $K = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{H}^{-1}\boldsymbol{x} - \boldsymbol{H}^{-1}\boldsymbol{a}\|_2 \le 1\} = \{\boldsymbol{a} + \boldsymbol{H}\boldsymbol{y} \mid \|\boldsymbol{y}\|_2 \le 1\}$ for a non-singular matrix $\boldsymbol{H} \in \mathbb{R}^{n \times n}$. Now consider the lattice $\Lambda := \Lambda(\boldsymbol{H}^{-1})$ and the ball $\mathcal{B}(\boldsymbol{H}^{-1}\boldsymbol{a}, 1)$ and apply Theorem 2.5.

- *Case (A):* If we find a lattice point in $\Lambda \cap \mathcal{B}(\boldsymbol{H}^{-1}\boldsymbol{a}, 1)$ then it will be of the form $\boldsymbol{H}^{-1}\boldsymbol{x}^*$ with $\boldsymbol{x}^* \in \mathbb{Z}^n$ and $\|\boldsymbol{H}^{-1}\boldsymbol{x}^* - \boldsymbol{H}^{-1}\boldsymbol{a}\|_2 \le 1$ and $\boldsymbol{x}^* \in K \cap \mathbb{Z}^n$ is exactly what we are looking for.

- *Case (B):* Suppose we find a dual lattice vector $\boldsymbol{c}^* \in \Lambda^*$ of length $\|\boldsymbol{c}^*\|_2 \leq 2^{O(n^2)}$. We know from Lemma 1.20 that the dual lattice basis is $((\boldsymbol{H}^{-1})^{-1})^T = \boldsymbol{H}^T$. Hence $\boldsymbol{c}^* = \boldsymbol{H}^T\boldsymbol{c}$ for some integer vector $\boldsymbol{c} \in \mathbb{Z}^n$. Then the integer width of that vector $\boldsymbol{c}$ is

$$\begin{aligned} w_{\boldsymbol{c}}(K) &= \max\{\boldsymbol{c}^T(\boldsymbol{x}-\boldsymbol{x}') \mid \boldsymbol{x},\boldsymbol{x}' \in K\} = \max\{\boldsymbol{c}^T(\boldsymbol{H}\boldsymbol{y}-\boldsymbol{H}\boldsymbol{y}') \mid \|\boldsymbol{y}\|_2, \|\boldsymbol{y}'\|_2 \leq 1\} \\ &= \max\{(\underbrace{\boldsymbol{H}^T\boldsymbol{c}}_{=\boldsymbol{c}^*})^T(\boldsymbol{y}-\boldsymbol{y}') \mid \|\boldsymbol{y}\|_2, \|\boldsymbol{y}'\|_2 \leq 1\} \leq 2 \cdot \|\boldsymbol{c}^*\|_2 \leq 2^{O(n^2)}. \end{aligned}$$

□

Now we have everything we need for the proof of the Flatness Theorem:

Proof of Khinchine's Flatness Theorem. Take $K \subseteq \mathbb{R}^n$ be any full-dimensional polytope[2]. We can find in polynomial time an ellipsoid $\mathcal{E}$ so that $\boldsymbol{c}+\mathcal{E} \subseteq K \subseteq \boldsymbol{c}+n\mathcal{E}$. Note that after a linear transformation, the Cor. 2.6 says that one can find either an integer point in that ellipsoid $\boldsymbol{c}+\mathcal{E}$ or one can find a direction so that at most $2^{O(n^2)}$ many hyperplanes intersect $\boldsymbol{c}+\mathcal{E}$. Then at most $n \cdot 2^{n^2/2}+1$ many hyperplanes will intersect $\boldsymbol{c}+n\cdot\mathcal{E} \supseteq K$. □

2.3 The algorithm

We can now give the complete algorithm:

Lenstra's algorithm for Integer Programming

Input: Polytope $K = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ given by matrix $\boldsymbol{A} \in \mathbb{Q}^{m\times n}$ and vector $\boldsymbol{b} \in \mathbb{Q}^m$.
Output: Either a point $K \cap \mathbb{Z}^n$ or decision that none exists.

(1) Compute an ellipsoid $\boldsymbol{a}+\mathcal{E} \subseteq K \subseteq \boldsymbol{a}+n\mathcal{E}$.
(2) Run the LLL-algorithm to find a basis $\boldsymbol{B} = (\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n)$ so that the orthogonality defect with respect to norm $\|\cdot\|_{\mathcal{E}}$ and inner product $\langle\cdot,\cdot\rangle_{\mathcal{E}}$ is bounded and $\|\boldsymbol{b}_1\|_{\mathcal{E}} \leq \ldots \leq \|\boldsymbol{b}_n\|_{\mathcal{E}}$.
(3) Write $\boldsymbol{a} = \sum_{i=1}^n \lambda_i \boldsymbol{b}_i$ with $\lambda_i \in \mathbb{R}$.
(4) IF $\boldsymbol{x}^* := (\sum_{i=1}^n \lfloor\lambda_i\rfloor \boldsymbol{b}_i) \in K$ THEN return $\boldsymbol{x}^*$
(5) Set $\boldsymbol{c} := \boldsymbol{b}_n^*$.
(6) FOR all $\delta \in \{\lceil\min\{\boldsymbol{c}\boldsymbol{x} \mid \boldsymbol{x} \in K\}\rceil,\ldots,\lfloor\max\{\boldsymbol{c}\boldsymbol{x} \mid \boldsymbol{x} \in K\}\rfloor\}$ DO
  (7) Run the Hermite normal form algorithm to find a lattice basis $\boldsymbol{B}' \in \mathbb{Q}^{n\times(n-1)}$ and an offset $\boldsymbol{d}$ with $\boldsymbol{d}+\Lambda(\boldsymbol{B}') = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \boldsymbol{c}\boldsymbol{x} = \delta\}$
  (8) Run the algorithm recursively to find integer point in $\{\boldsymbol{x}' \in \mathbb{R}^{n-1} \mid \boldsymbol{A}(\boldsymbol{d}+\boldsymbol{B}'\boldsymbol{x}') \leq \boldsymbol{b}\}$
(9) Return any point in $K \cap \mathbb{Z}^n$ that was found or decide that none exists otherwise

Theorem 2.7. *For a polytope $K = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$, Lenstra's algorithm finds a point $\boldsymbol{x}^* \in K \cap \mathbb{Z}^n$ in time $2^{O(n^3)}$ times a polynomial in the encoding length of $\boldsymbol{A}$ and $\boldsymbol{b}$ (if there is any such point).*

Proof. Suppose that $T(n)$ be the number of recursive calls that are made to the algorithm to solve an instance of dimension n. Then the total running time will be $T(n)$ times a polynomial factors. We have proven that we can split each subproblem into $2n \cdot 2^{n^2/2}+1$ many subproblems. Thus $T(n) \leq T(n-1)\cdot(2n\cdot 2^{n^2}+1)$. Reiterating this gives $T(n) \leq \prod_{k=1}^n (2k\cdot 2^{k^2/2}+1) \leq 2^{O(n^3)}$. That shows the claim. □

[2] We assume that the polytope K is defined by a rational inequality system — if K is not fulldimensional, then we can directly eliminate one dimension.

The currently best known algorithm for integer programming takes time $n^{O(n)}$ times a polynomial in the input length and is due to Kannan [Kan83]. In each iteration, Kannan's algorithm spends time $n^{O(n)}$ to find a "more orthogonal" basis which then allows to recurse on only poly(n) many $(n-1)$-dimensional subproblems.

Open Question 2. It is a big open question whether one can bring the running time down to single-exponential, that means whether one can replace $n^{O(n)}$ by $2^{O(n)}$. This would somewhat unify algorithms for a couple of problems such as Shortest Vector and Closest Vector that we will study in detail in Chapter 4 and 5. In particular if one stares at Lenstra's algorithm then the most obvious waste of resources is that all the slices are actually slices of the *same* polytope with translates of the *same* hyperplane. One might wonder whether it is possible to do some preprocessing once instead of treating all slices separately.

2.4 Exercises

Exercise 2.1. Let $K \subseteq \mathbb{R}^n$ be a convex set. While we know by John's Theorem that there is an n-approximate ellipsoid for K, we only know how to find an $O(n^{3/2})$-approximate ellipsoid in polynomial time. In this exercise, we want to discuss how. So suppose for the sake of simplicity that K is so transformed that the unit ball $\mathcal{B}(\mathbf{0},1)$ is contained in K. Moreover, suppose that K is *not* contained in $C'n^{3/2}\cdot K$, where $C'>0$ is a large enough constant. Then prove that one can find an ellipsoid $\mathcal{E} \subseteq K$ in polynomial time that has $\mathrm{vol}(\mathcal{E}) \geq 2\cdot\mathrm{vol}(K)$.
Remark: For the convex set K you can assume to have a polynomial time algorithm for the separation problem and for optimizing a linear function over K.

Exercise 2.2. Let $\|\cdot\|_*$ be any norm in $\mathbb{R}^n$. Show that there is a matrix $\boldsymbol{A}$ so that $\|\boldsymbol{Ax}\|_2 \leq \|\boldsymbol{x}\|_* \leq \sqrt{n}\|\boldsymbol{Ax}\|_2$.
Remark: The exercise shows that in $\mathbb{R}^n$ after a linear transformation all norms are equivalent to the Euclidean norm up to a factor of $\sqrt{n}$. The worst case is attained for $\|\cdot\|_1$ and $\|\cdot\|_\infty$ that both are a factor $\sqrt{n}$ away from the Euclidean norm (and in fact they are a factor of n away from each other).

Exercise 2.3. Consider $K := \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}-\boldsymbol{a}\|_2 \leq \frac{1}{4}\sqrt{n}\}$ where $\boldsymbol{a} := (\frac{1}{2},\ldots,\frac{1}{2})$. Show that $K \cap \mathbb{Z}^n = \emptyset$. Then give $w(K)$ (up to a constant factor).

Exercise 2.4. Let $K \subseteq \mathbb{R}^n$ be a symmetric convex body. Show that there is an ellipsoid $\mathcal{E}$ (with the origin as center) so that $E \subseteq K \subseteq C\sqrt{n}\cdot K$ for some (large) constant $C>0$.
Remark: You can modify the proof from the lecture that gives a n-approximation for arbitrary convex sets.

Exercise 2.5. Show that for any lattice $\Lambda \subseteq \mathbb{R}^n$ one has $\lambda_n(\Lambda)\cdot\lambda_1(\Lambda^*) \leq 2^{O(n^2)}$.
Remark: A stronger claim of [Ban93a] shows that $\lambda_n(\Lambda)\cdot\lambda_1(\Lambda^*) \leq O(n)$. But the weaker bound I'm asking here follows already from the material that we saw in the lectures.

Exercise 2.6. Recall that the simultaneuous diophantine approximation problem is as follows: Given rational numbers $\alpha_1,\ldots,\alpha_d \in [0,1]$ and a bound $Q \in \mathbb{N}$, find the denominator $q \in \{1,\ldots,Q\}$ and nominators $p_1,\ldots,p_d \in \mathbb{Z}$ so that

$$\max_{i=1,\ldots,d} |q\alpha_i - p_i|$$

is minimized. Show that if d is a fixed constant, this problem can be solved in polynomial time.

Exercise 2.7. In this exercise, we want to discuss a constructive version of Minkowski's Theorem using John's theorem and the LLL-algorithm. Let $K \subseteq \mathbb{R}^n$ be a fulldimensional, symmetric convex body with $\mathrm{vol}(K) \geq 1$. Show that one can compute a vector $\boldsymbol{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$ with $\boldsymbol{x} \in 2^{O(n)} \cdot K$ in polynomial time.

Exercise 2.8. Formulate and prove a constructive version of Dirichlet's theorem (in other words, find a non-trival approximation in polynomial time).

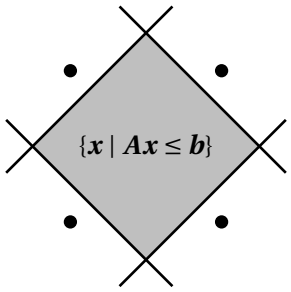
Chapter 3

Two Structural Theorems in Integer optimization

Even though integer programming is an **NP**-hard problem, it still allows some structural insights that we want to discuss in this chapter.

3.1 Doignon's Theorem

For the first result, we want to start consider a linear program $\boldsymbol{Ax} \leq \boldsymbol{b}$. It is well-known that if this system is infeasible — that means $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} \leq \boldsymbol{b}\} = \emptyset$ — then there is actually a subsystem of at most $n+1$ constraints that is already infeasible. We will see the proof in the exercises. But for now let us discuss the analogue for *integer linear programs*. Here the situation is definitely different. For example in $\mathbb{R}^2$ we can easily create an integer-infeasible system $\boldsymbol{Ax} \leq \boldsymbol{b}$ with 4 constraints so that the removal of any constraints would make it feasible:



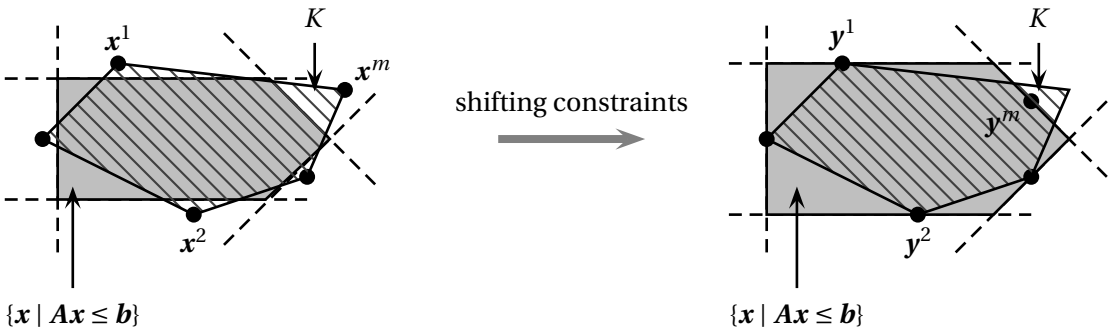
That brings us to Doignon's Theorem [Doi73] that tells us that this situation is tight in 2-dimensions. We want to mention that this theorem had been rediscovered later by Bell [Bel77] and Scarf [Sca77]. Sometimes the theorem is also called Doignon-Bell-Scarf Theorem. In our exposition, we follow [Sch99].

Theorem 3.1 (Doignon's Theorem 1973)**.** *Let $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} \leq \boldsymbol{b}\}$ be a system with $P \cap \mathbb{Z}^n = \emptyset$. Then there is a subsystem of at most 2^n inequalities that are already integer-infeasible.*

Proof. Suppose that $\boldsymbol{Ax} \leq \boldsymbol{b}$ is a minimal integer infeasible constraint system in the sense that after the removal of any constraint, the system admits an integral solution. Let m be the number of constraints. We know by assumption that there are points $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m \in \mathbb{Z}^n$ so that $\boldsymbol{x}^i$ satisfies all constraints except the ith one. Let $K := \operatorname{conv}\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m\}$. Now repeat the following process: Take any constraint i so that

no point in $K \cap \mathbb{Z}^n$ lies in the interior of face i (meaning that no integer point in K lies on $\boldsymbol{A}_i\boldsymbol{x} = \boldsymbol{b}_i$ but satisfies all other constraints strictly). Then increase its right hand side until at least one point in $K \cap \mathbb{Z}^n$ lies in its interior. This process will stop at some point since the set of points $K \cap \mathbb{Z}^n$ is finite. We also note that it is not possible that a right hand side $\boldsymbol{b}_i$ is pushed to ∞ since there is the point $\boldsymbol{x}^i$ that is only violated by the ith constraint.

Now we have a sequence of integer points $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^m \in K \cap \mathbb{Z}^n$ (these may or may not be the original points $\boldsymbol{x}^i$) so that $\boldsymbol{A}_i\boldsymbol{y}^i = \boldsymbol{b}_i$ and $\boldsymbol{A}_i\boldsymbol{y}^j < \boldsymbol{b}_i$ for all $i \neq j$. Note that there is no point in $K \cap \mathbb{Z}^n$ with $\boldsymbol{A}\boldsymbol{x} < \boldsymbol{b}$[1]. If $m > 2^n$, then there must be two of those points that have the same parity. Say these are $\boldsymbol{y}^1$ and $\boldsymbol{y}^2$ and having the same parity means that $y^1_k \equiv_2 y^2_k$ for each coordinate $k \in \{1, \ldots, n\}$. In particular the midpoint $\boldsymbol{y} := \frac{1}{2}(\boldsymbol{y}^1 + \boldsymbol{y}^2)$ is integral and $\boldsymbol{A}\boldsymbol{y} < \boldsymbol{b}$ (well, and $\boldsymbol{y} \in K \cap \mathbb{Z}^n$). That gives a contradiction.



□

There is also a slightly more general form of the theorem.

Theorem 3.2 (General Form of Doignon's Theorem 1973)**.** *Let $K_1, \ldots, K_m \subseteq \mathbb{R}^n$ be convex sets so that the intersection $K_1 \cap \ldots \cap K_m$ does not contain an integer point. Then there is a subset $I \subseteq \{1, \ldots, m\}$ of size $|I| \leq 2^n$ so that already $\bigcap_{i \in I} K_i$ does not contain an integral point.*

Note that the general version implies the version that we have proven by setting $K_i := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}_i\boldsymbol{x} \leq \boldsymbol{b}_i\}$ as the ith half-space.

3.2 Complexity of the integer hull

Let $\boldsymbol{P} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ be a polytope (though much of our discussion makes also sense for convex bodies). Recall that a point $\boldsymbol{x} \in P$ is called a *vertex / extreme point* if there is no $\boldsymbol{d} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ so that $\{\boldsymbol{x} + \boldsymbol{d}, \boldsymbol{x} - \boldsymbol{d}\} \subseteq P$. Then a particularly useful set is the *convex hull of the integer points*

$$P_I := \text{conv}(P \cap \mathbb{Z}^n)$$

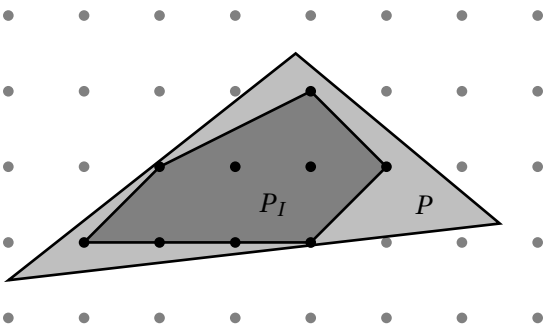
or a bit shorter, the *integer hull*. The reason is that if we are interested in the integer linear program $\max\{\boldsymbol{c}\boldsymbol{x} \mid \boldsymbol{x} \in P \cap \mathbb{Z}^n\}$, then convexifying the solution set does not change the optimum value, that means

$$\max\{\boldsymbol{c}\boldsymbol{x} \mid \boldsymbol{x} \in P \cap \mathbb{Z}^n\} = \max\{\boldsymbol{c}\boldsymbol{x} \mid \boldsymbol{x} \in P_I\}$$

as we are optimizing a linear function. Note that as we are assuming that P is bounded, the optimum of this linear program is always attained at a vertex (it might be though that the optimum solution is not

[1] But we have not excluded the possibility that some other integer point now satisfies $\boldsymbol{A}\boldsymbol{x} < \boldsymbol{b}$

unique — then a whole face of P_I will be optimal, but that face also contains some optimal vertices). In other words, even if P_I contains a huge number of integer points, only those that are vertices will appear as possible solutions to integer linear programs. The question that we want to answer in this section is: *how many vertices can P_I have?*



From the above example we already see that the number of vertices of P_I might actually be larger than the ones of P (and vice versa). We will even see in the exercises that the number of vertices cannot be bounded by a function on n. In particular for any k one can find a polytope in $\mathbb{R}^2$ with 3 constraints and at least k vertices. Somewhat surprisingly, if the dimension is fixed, the number of vertices of the integer hull is bounded by a polynomial in the encoding length of the polytope.

Here is an example application of this fact: we have seen that integer linear programs in fixed dimension can be solved in polynomial time using Lenstra's algorithm. Imagine we wanted to solve an integer linear program without knowing the objective function apriori. Still, in a preprocessing step one could compute find the polynomial size list of extreme points. If then the objective function arrives one simply has to select the best of those points with respect to the objective function.

Theorem 3.3 ([Har88, CHKM92])**.** *Let $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ be a polytope. Assume that $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{Z}^m$ and $\|\boldsymbol{A}\|_\infty, \|\boldsymbol{b}\|_\infty \leq \Delta$ and $P \subseteq [-\Delta, \Delta]^n$. Then the number of vertices of P_I is bounded by $(O(\log(n\Delta)))^m$.*

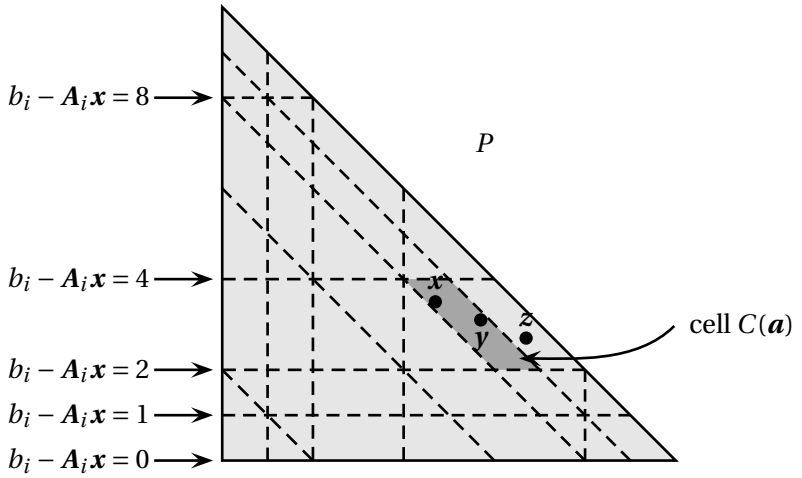
Proof. For $\alpha \in \mathbb{Z}_{\geq 0}$, let us define values

$$L(\alpha) := \begin{cases} 2^{\alpha-1} & \text{if } \alpha \geq 1 \\ 0 & \text{if } \alpha = 0 \end{cases} \qquad \text{and} \qquad U(\alpha) := \begin{cases} 2^{\alpha} & \text{if } \alpha \geq 1 \\ 0 & \text{if } \alpha = 0 \end{cases}.$$

For a vector $\boldsymbol{a} \in \mathbb{Z}^m_{\geq 0}$ we define *cells*

$$C(\boldsymbol{a}) := \{\boldsymbol{x} \in \mathbb{R}^n \mid L(a_i) \leq b_i - \boldsymbol{A}_i\boldsymbol{x} \leq U(a_i)\ \forall i \in [m]\}$$

The cells cover all the integer points in P, that means $P \cap \mathbb{Z}^n \subseteq \bigcup_{\boldsymbol{a}\in\mathbb{Z}^m_{\geq 0}} C(\boldsymbol{a})$. Note that the cells do not actually cover all *fractional* points, but all vertices of P_I must be integral, so they are definitely covered.



Next, let us count the number of non-empty cells. Fix a point $\boldsymbol{x} \in P$ and a constraint i. Then the slack of the point is at most $|\boldsymbol{A}_i\boldsymbol{x} - b_i| \leq (n+1)\Delta^2$ using that $\|\boldsymbol{A}\|_\infty, \|\boldsymbol{b}\|_\infty, \|\boldsymbol{x}\|_\infty \leq \Delta$. Then we only need to consider cells generated by $\boldsymbol{a} \in \mathbb{Z}^m_{\geq 0}$ with $\|\boldsymbol{a}\|_\infty \leq \log_2((n+1)\Delta^2) + 2$. Hence, there are at most $(O(\log_2(n\Delta))^m$ cells. The remaining part of the argument is to bound the vertices per cell:

Claim. Each cell $C(\boldsymbol{a})$ contains at most one vertex of P_I.

Proof. Consider two vertices $\boldsymbol{x}, \boldsymbol{y} \in \text{vert}(P_I) \cap C(\boldsymbol{a})$ (see the figure above). Note that $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^n$. The crucial properties of the cells is that the points $\boldsymbol{x}, \boldsymbol{y}$ have roughly the same slack for all constraints (up to a factor of 2). Now we define $\boldsymbol{z} := \boldsymbol{x} + 2 \cdot (\boldsymbol{y} - \boldsymbol{x}) = 2\boldsymbol{y} - \boldsymbol{x}$. We claim that $\boldsymbol{z} \in P$. So, fix a constraint i. If $a_i = 0$, then $b_i - \boldsymbol{A}_i\boldsymbol{z} = 0$. Now suppose that $a_i \geq 1$. Then

$$b_i - \boldsymbol{A}_i\boldsymbol{z} \geq \underbrace{b_i - \boldsymbol{A}_i\boldsymbol{y}}_{\geq 2^{a_i-1}} - \underbrace{|\boldsymbol{A}_i(\boldsymbol{y} - \boldsymbol{x})|}_{\leq 2^{a_i} - 2^{a_i-1}} \geq 0.$$

Thus $\boldsymbol{z} \in P$. By construction $\boldsymbol{z} \in \mathbb{Z}^n$ and hence $\boldsymbol{z} \in P_I$. That implies that $\boldsymbol{y}$ was on the line segment between $\boldsymbol{x}$ and $\boldsymbol{z}$ and we get a contradiction to $\boldsymbol{y}$ being an extreme point. □

Overall, every cell contains at most one extreme point of P_I, so the claim follows. □

Note that there is a better bound if $m \gg n$: We can consider the cells as a *hyperplane arrangement* in $\mathbb{R}^n$ with $k := O(m\log(n\Delta))$ many hyperplanes. A well-known theorem in discrete geometry says that the number of cells is bounded by $O(k^n)$. These arguments can be found in the book of Matousek [Mat02]. Again, since each cell contains only one integer point, this gives an improved bound of $m^n \cdot (O(\log(n\Delta)))^n$.

3.3 Exercises

Exercise 3.1. For any n, give a linear inequality system $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ with 2^n constraints so that there is no $\boldsymbol{x} \in \mathbb{Z}^n$ with $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ while removing any constraint makes it integer feasible.

Exercise 3.2. We want to prove the more general version of Doignon's Theorem[2]: Suppose that $K_1, \ldots, K_m \subseteq \mathbb{R}^n$ are convex and bounded and $(\bigcap_{i=1}^m K_i) \cap \mathbb{Z}^n = \emptyset$. Then there is a subset $I \subseteq [m]$ of size $|I| \leq 2^n$ so that $(\bigcap_{i \in I} K_i) \cap \mathbb{Z}^n = \emptyset$.

Remark: You are allowed to use the version that we have proven in the lecture as a blackbox.

[2]Well, I have added the boundedness to avoid messy arguments

Exercise 3.3. Consider the irrational polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^2 \mid x_1 \geq 1;\ x_2 \geq 0; x_2 \leq \sqrt{2} \cdot x_1\}$. Show that P_I has infinitely many vertices[3].

Exercise 3.4. Suppose that $P = \{\boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\} \subseteq \mathbb{R}^m$ is a polytope with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$. Let $\text{int}(P) := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} < \boldsymbol{b}\}$ be the *interior*. Suppose that on each facet i there is a point $\boldsymbol{x}^i \in P \cap \mathbb{Z}^n$ with $\boldsymbol{A}_i \boldsymbol{x}^i = b_i$ but $\boldsymbol{A}_j \boldsymbol{x}^i < b_j$ for $j \neq i$. Moreover suppose that $k := |\text{int}(P) \cap \mathbb{Z}^n|$ and that after removing any constraint from P, the number of integer points in the interior would exceed k. Prove that $m < (k+2) \cdot 2^n$.

[3] Of course, one could alternatively truncate P it to a polytope $P' = P \cap \{\boldsymbol{x} \mid x_1 \leq k\}$ and then argue that the number of vertices of P'_I grows with k.

Chapter 4

A $2^{O(n)}$-time algorithm for the Shortest Vector Problem

Recall that the *Shortest Vector Problem* (SVP) that we have seen earlier in Chapter 1:

> SHORTEST VECTOR PROBLEM (SVP)
> **Input:** A lattice $\Lambda(\boldsymbol{B})$ given by a regular matrix $\boldsymbol{B} \in \mathbb{Q}^{n\times n}$.
> **Goal:** Find the vector $\boldsymbol{x} \in \Lambda(\boldsymbol{B})/\{\boldsymbol{0}\}$ minimizing $\|\boldsymbol{x}\|_2$.

Here, we want to describe the elegant algorithm of Ajtai, Kumar and Sivakumar von 2001 [AKS01], which finds the exact Shortest Vector in time $2^{O(n)} \cdot \text{poly(input)}$. Previously, the best known algorithm was a $n^{O(n)} \cdot \text{poly(input)}$ algorithm by Kannan [Kan87a]. For this chapter, we will follow the exposition in the excellent lecture notes of Regev [Reg09][1]. Recall that SVP is **NP**-hard, so it is not surprising that an exponential running time is needed. By a slight abuse of notation, we denote $\text{SVP}(\Lambda(\boldsymbol{B}))$ as the length of the shortest vector.

It is a standard argument, that one can assume to know the length of the shortest vector up to some small factor.

Lemma 4.1. *It suffices to have an* SVP *algorithm for instances with* $2 \le \text{SVP}(\Lambda(\boldsymbol{B})) < 3$.

Proof. There is an integer $k \in \mathbb{Z}$ so that the scaled lattice $\Lambda(1.5^k \boldsymbol{B})$ has its shortest vector in the interval $[2,3[$. We can use the LLL-algorithm to find an estimate $\Delta \in \mathbb{Q}$ so that $\Delta \le \text{SVP}(\Lambda(\boldsymbol{B})) \le 2^{O(n)} \cdot \Delta$. Then there are only $O(n)$ many options for the correct choice of k. We simply try out all of them; running the actual algorithm $O(n)$ times. For all vectors that are found, we can check whether they are in $\Lambda(\boldsymbol{B})$ and then return the shortest non-zero vector. □

From now on, we make the assumption that $2 \le \text{SVP}(\Lambda(\boldsymbol{B})) \le 3$.

4.1 Packing balls in $\mathbb{R}^n$

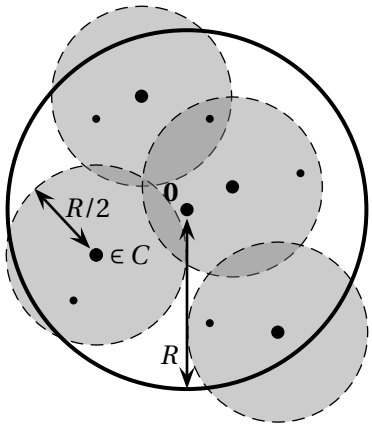
Next, we want to introduce some basic geometric facts about $\mathbb{R}^n$ that make the whole argument work. Let $\mathcal{B}(\boldsymbol{0}, R)$ be the *ball of radius R* with the origin as center.

[1] See also `http://www.cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/svpalg.pdf`

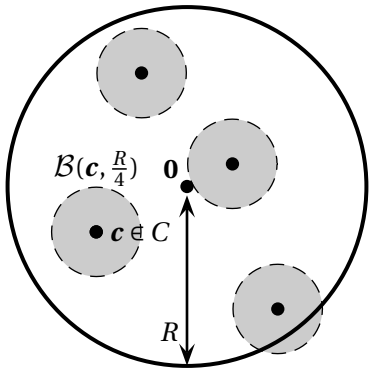
Lemma 4.2. *If we have more than 5^n many points in $\mathcal{B}(\mathbf{0}, R)$, then some of them will have distance at most $\frac{R}{2}$.*

Instead of proving this, let us immediately prove something slightly more general. We will use $d(\boldsymbol{x}, C) := \min\{\|\boldsymbol{c} - \boldsymbol{x}\|_2 \mid \boldsymbol{c} \in C\}$ as the Euclidean distance of $\boldsymbol{x}$ to the nearest point in C.

Lemma 4.3. *Let $X \subseteq \mathcal{B}(\mathbf{0}, R) \subseteq \mathbb{R}^n$ be a finite set of points. Then one can find a subset of* centers *$C \subseteq X$ with $|C| \leq 5^n$ so that $d(\boldsymbol{x}, C) \leq \frac{R}{2}$ for all $\boldsymbol{x} \in X$.*



Proof. Starting with $C := \emptyset$, greedily add a point from X to C if it has a distance bigger than $R/2$ to all other points that are currently in C. Trivially, this way we obtain a set of clusters so that $d(\boldsymbol{x}, C) \leq R/2$ for all $\boldsymbol{x} \in X$. It remains to argue that $|C| \leq 5^n$. Observe that for different centers $\boldsymbol{c}, \boldsymbol{c}' \in C$ we have $\|\boldsymbol{c} - \boldsymbol{c}'\|_2 > R/2$ (since otherwise, we would not have added one of those centers). That means that the balls $\mathcal{B}(\boldsymbol{c}, \frac{R}{4})$ for all centers $\boldsymbol{c} \in C$ do not overlapp.



Observe that all those balls $\mathcal{B}(c, \frac{R}{4})$ are fully contained in $\mathcal{B}(0, \frac{5}{4}R)$. Inspecting the volume we see that

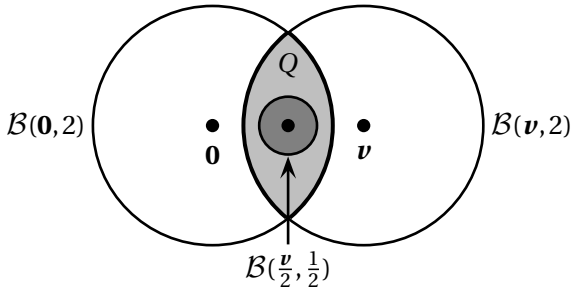
$$\frac{\text{vol}(\mathcal{B}(\mathbf{0}, \frac{5}{4}R))}{\text{vol}(\mathcal{B}(\mathbf{0}, \frac{R}{4}))} = 5^n$$

In other words, there is no way that we could pack more than 5^n balls of radius $\frac{R}{4}$ into a ball of radius $\frac{5}{4}R$. Hence $|C| \leq 5^n$. □

Observe that the clustering procedure runs in time polynomial in $|X|$. There are two more volume insights that we will need later in the analysis.

Lemma 4.4. *Let $\boldsymbol{v} \in \mathbb{R}^n$ be a vector of length $2 \le \|\boldsymbol{v}\|_2 \le 3$. Let $Q := \mathcal{B}(\mathbf{0},2) \cap \mathcal{B}(\boldsymbol{v},2)$. Then*

$$\frac{vol(Q)}{vol(\mathcal{B}(\mathbf{0},2))} \ge 2^{-2n}$$



Proof. The set Q contains a ball of radius $\frac{1}{2}$ and center $\frac{\boldsymbol{v}}{2}$, hence the volume ratio is at most 4^n. □

Finally we need the insight that there are not too many short vectors in a lattice.

Lemma 4.5. *Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice with $\text{SVP}(\Lambda) \ge 2$. Then $|\Lambda \cap \mathcal{B}(\mathbf{0},8)| \le 2^{4n}$.*

Proof. If the shortest lattice vector has length at least 2, then we can put a ball of radius 1 around each lattice point in $\Lambda \cap \mathcal{B}(\mathbf{0},8)$ and those balls will not overlap. Since all those balls will be contained in $\mathcal{B}(\mathbf{0},9)$, their number cannot exceed $9^n \le 2^{4n}$. □

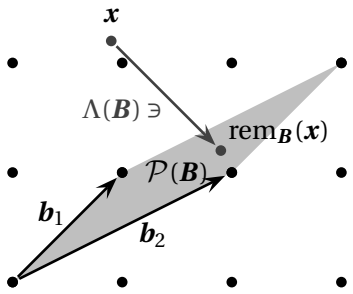
4.2 The basic approach

The idea behind the sieve algorithm is as follows: we sample an exponential number of points $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N \sim \mathcal{B}(\mathbf{0},2)$ from a ball of radius 2. We use those points to compute some (fairly long) lattice vectors $\boldsymbol{z}_i \in \Lambda(\boldsymbol{B})$. Starting with those points, the algorithm we will maintain a list $(\boldsymbol{x}_1,\boldsymbol{z}_1),\ldots,(\boldsymbol{x}_N,\boldsymbol{z}_N)$ satisfy the following invariants:

- *Invariant 1:* $\boldsymbol{z}_i \in \Lambda(\boldsymbol{B})$ for all $i = 1,\ldots,N$
- *Invariant 2:* $\|\boldsymbol{y}_i\|_2 \le R$ for all $i = 1,\ldots,N$ where $\boldsymbol{y}_i := \boldsymbol{x}_i + \boldsymbol{z}_i$

At the beginning, this is satisfied for some huge number R. Then in each iteration we will substract $\boldsymbol{z}_i$'s from each other so that we obtain shorter and shorter lattice vectors. In fact, in each iteration we will be able to decrease the parameter R by a factor of roughly 2. Somewhat unintuitively, we will *not* change the $\boldsymbol{x}_i$'s — they will only be helpful in the analysis to guarantee that we find a short non-zero vector.

We should begin with discussing how the list can be "initialized". In other words, how can we find such a list for at least for some terribly large value of R? Recall that for a basis $\boldsymbol{B} \in \mathbb{Q}^{n\times n}$, the *fundamental parallelepiped* is $\mathcal{P}(\boldsymbol{B}) := \{\sum_{i=1}^n \lambda_i \boldsymbol{b}_i \mid 0 \le \lambda_i < 1 \ \forall i \in [n]\}$. For a point $\boldsymbol{x} \in \mathbb{R}^n$ we define the *remainder* as the unique vector $\text{rem}_{\boldsymbol{B}}(\boldsymbol{x}) \in \mathcal{P}(\boldsymbol{B})$ so that $\text{rem}_{\boldsymbol{B}}(\boldsymbol{x}) - \boldsymbol{x} \in \Lambda(B)$. Phrased differently, starting at $\boldsymbol{x}$, we can add a lattice vector and "jump" into the fundamental parallelepiped. The translated point that we get will be $\text{rem}_{\boldsymbol{B}}(\boldsymbol{x})$. For example for the standard lattice $\mathbb{Z}^2$ with the basis $(1,1),(2,1)$, the fundamental parallelepiped and $\text{rem}_{\boldsymbol{B}}(\boldsymbol{x})$ look as follows:



Lemma 4.6. *For any point $\boldsymbol{x} \in \mathbb{R}^n$, the remainder $\mathrm{rem}_{\boldsymbol{B}}(\boldsymbol{x})$ can be computed in poly-time.*

Proof. Since the matrix $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is regular, there is a unique linear combination

$$\boldsymbol{x} = \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i.$$

We can use Gaussian elimination to find the coefficient vector $\boldsymbol{\lambda} \in \mathbb{R}^n$. Then

$$\mathrm{rem}_{\boldsymbol{B}}(\boldsymbol{x}) = \sum_{i=1}^{n} (\lambda_i - \lfloor \lambda_i \rfloor) \boldsymbol{b}_i$$

is the remainder. □

We can use this observation to generate a list of (admittedly fairly long) lattice vectors:

Initialization

(1) Set $R_0 := n \cdot \max_{i=1,\ldots,n} \|\boldsymbol{b}_i\|_2$ where $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ are the columns of B.
(2) For $N := 2^{8n} \log(R_0)$, sample independently random points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathcal{B}(\boldsymbol{0}, 2)$.
(3) Compute $\boldsymbol{z}_i := \mathrm{rem}_B(\boldsymbol{x}_i) - \boldsymbol{x}_i \in \Lambda(\boldsymbol{B})$ for $i = 1, \ldots, N$

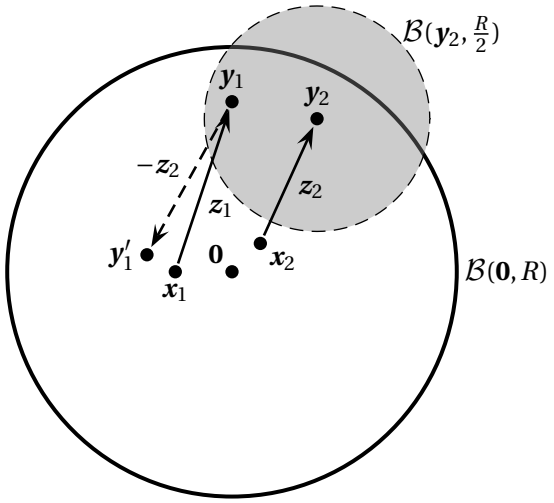
Let us verify that the obtained list satisfies the invariants for parameter R_0:

- *Invariant 1:* The difference between $\boldsymbol{x}_i$ and the remainder $\mathrm{rem}_{\boldsymbol{B}}(\boldsymbol{x}_i)$ is always a lattice point, hence $\boldsymbol{z}_i \in \Lambda(\boldsymbol{B})$.
- *Invariant 2:* By definition $\boldsymbol{y}_i = \mathrm{rem}_B(\boldsymbol{x}_i) \in \mathcal{P}(\boldsymbol{B})$, hence

$$\|\boldsymbol{y}_i\|_2 = \|\mathrm{rem}_{\boldsymbol{B}}(\boldsymbol{x}_i)\|_2 \leq \sum_{j=1}^{n} \|\boldsymbol{b}_j\|_2 \leq R_0.$$

4.3 The main sieving procedure

In the main algorithm we will iteratively subtract the lattice vectors from each other in order to obtain shorter lattice vectors. To make the principle clear, suppose we are in the middle of our algorithm with a parameter R that is somewhere between 2 and R_0. We consider our list $(\boldsymbol{x}_1, \boldsymbol{z}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{z}_N)$ and suppose that that $\|\boldsymbol{y}_i\|_2 \leq R$ for all i. Since we have $N \gg 5^n$, there must be some pairs, say $(\boldsymbol{x}_1, \boldsymbol{z}_1)$ and $(\boldsymbol{x}_2, \boldsymbol{z}_2)$ so that $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2 \leq \frac{R}{2}$. This also implies that $\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_2 \leq \frac{R}{2} + 4$. Hence if we subtract one from another, say $\boldsymbol{z}_1' := \boldsymbol{z}_1 - \boldsymbol{z}_2$, then the length of $\boldsymbol{z}_1$ and $\boldsymbol{y}_1$ is getting a lot shorter.



But we also see a potential danger: if $\boldsymbol{z}_1 = \boldsymbol{z}_2$ then the new lattice vector for the 1st pair is $z_1' = \boldsymbol{0}$, which is a very short vector, but it is also completely useless. The trick in the algorithm is to make sure that we will have at least some short vectors at the end that are not $\boldsymbol{0}$.

The complete sieving algorithm is as follows:

Sieve algorithm

Input: A lattice basis $\boldsymbol{B} \in \mathbb{R}^{n \times n}$

Output: The shortest vector in the lattice $\Lambda(\boldsymbol{B})$

(1) Initialize a list $Z = (\boldsymbol{x}_1, \boldsymbol{z}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{z}_N)$ satisfying both invariants for parameter $R := R_0$.

(2) WHILE $R > 6$ DO

  (3) Perform a clustering for $\boldsymbol{y}_i := \boldsymbol{x}_i + \boldsymbol{z}_i$ and let C be the cluster centers.
  Let $\sigma(i) \in C$ be the index so that $\|\boldsymbol{y}_i - \boldsymbol{y}_{\sigma(i)}\|_2 \leq \frac{R}{2}$

  (4) Delete the cluster centers from the list Z

  (5) For each remaining pair, set $\boldsymbol{z}_i := \boldsymbol{z}_i - \boldsymbol{z}_{\sigma(i)}$

  (6) Set $R := \frac{R}{2} + 2$.

(7) Return the shortest non-zero vector among all pairs $\boldsymbol{z}_i - \boldsymbol{z}_j$

First, let us observe that the two invariants are indeed maintained:

- *Invariant 1:* We always have $\boldsymbol{z}_i \in \Lambda(\boldsymbol{B})$ as we only subtract/add lattice vectors
- *Invariant 2:* After updating a vector $\boldsymbol{y}_i$ to $\boldsymbol{y}_i'$ we have

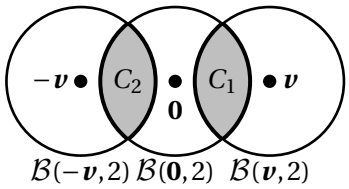
$$\|\boldsymbol{y}_i'\|_2 = \|\boldsymbol{y}_i - \boldsymbol{z}_{\sigma(i)}\|_2 \leq \underbrace{\|\boldsymbol{y}_i - \boldsymbol{y}_{\sigma(i)}\|_2}_{\leq R/2} + \underbrace{\|\boldsymbol{x}_{\sigma(i)}\|_2}_{\leq 2} \leq \frac{R}{2} + 2$$

Note that in principle it seems we have no mechanism that prevents that after a couple of iterations all lattice vectors are $\boldsymbol{z}_i = \boldsymbol{0}$. That's where the random starting points $\boldsymbol{x}_i$ come into play.

In the remaining part, we will prove that in step (7) we find the shortest vector with high probability. For the sake of the analysis, let $\boldsymbol{v} \in \Lambda(\boldsymbol{B})$ denote the shortest lattice vector. Let us define the regions

$$C_1 := \mathcal{B}(\boldsymbol{0}, 2) \cap \mathcal{B}(\boldsymbol{v}, 2) \quad \text{and} \quad C_2 := \mathcal{B}(\boldsymbol{0}, 2) \cap \mathcal{B}(-\boldsymbol{v}, 2)$$

that we visualize below:



We also define a bijective map $\tau : \mathbb{R}^n \to \mathbb{R}^n$ with

$$\tau(\boldsymbol{x}) = \begin{cases} \boldsymbol{x}+\boldsymbol{v} & \text{if } \boldsymbol{x} \in C_2 \\ \boldsymbol{x}-\boldsymbol{v} & \text{if } \boldsymbol{x} \in C_1 \\ \boldsymbol{x} & \text{otherwise} \end{cases}$$

In other words, the function τ maps points in C_1 to C_2 and vice versa (but does leave other points invariant). We call this "flipping". In the initialization step of the algorithm we choose the $\boldsymbol{x}_i$'s uniformly from $\mathcal{B}(\mathbf{0},2)$. Obviously that distribution is the same as the following:

(1) Pick $\boldsymbol{x}_i \sim \mathcal{B}(\mathbf{0},2)$ uniformly at random

(2) Tossing: With probability $\frac{1}{2}$, "flip" $\boldsymbol{x}_i$ (that means replace $\boldsymbol{x}_i$ with $\tau(\boldsymbol{x}_i)$)

The trick for the analysis of the algorithm is as follows: we imagine that we change the algorithm and run step (2) for a pair $(\boldsymbol{x}_i, \boldsymbol{z}_i)$ just before the first time that it actually mattered whether $\boldsymbol{x}_i$ will be flipped. This is not going to change the behavior of the algorithm (and as this is just for the sake of the analysis, it does not matter that the optimal vector $\boldsymbol{v}$ is used in generating the distribution).

Lemma 4.7. *With high probability, the shortest vector $\boldsymbol{v}$ is among the pairs $\boldsymbol{z}_i - \boldsymbol{z}_j$ for some surviving indices i, j.*

Proof. First note that after the initialization step we have $\boldsymbol{y}_i = \mathrm{rem}_B(\boldsymbol{x}_i)$. For $\boldsymbol{y}_i$ it does not matter whether $\boldsymbol{x}_i$ will be flipped or not, since $\mathrm{rem}_{\boldsymbol{B}}(\boldsymbol{x}) = \mathrm{rem}_{\boldsymbol{B}}(\boldsymbol{x}+\boldsymbol{v}) = \mathrm{rem}_{\boldsymbol{B}}(\boldsymbol{x}-\boldsymbol{v})$. On the other hand, the vector $\boldsymbol{z}_i$ will dependent on whether $\boldsymbol{x}_i$ is flipped or not. In other words, as long as the algorithm can work with $\boldsymbol{y}_i$ we do not yet need to decide whether $\boldsymbol{x}_i$ will be flipped. Only the first time that $\boldsymbol{z}_i$ is needed, this decision has to be made.

Now consider the first clustering step. For the clustering itself, only the $\boldsymbol{y}_i$'s are used, so we did not need to decide the flipping for the $\boldsymbol{x}_i$'s. But in the update step we set $\boldsymbol{z}_i := \boldsymbol{z}_i - \boldsymbol{z}_{\sigma(i)}$. Well, this update can be equivalently written as setting $\boldsymbol{y}_i := \boldsymbol{y}_i - \boldsymbol{z}_{\sigma(i)}$. In other words, the side of $\boldsymbol{x}_i$ does not yet matter, but the side of the cluster center $\sigma(i)$ does matter. Hence before doing this update step, we do need to decide the flipping of the cluster centers. But this is the reason why we delete the cluster centers after they have been used once: so that throughout the algorithm we only have pairs for which we did not need to decide whether or not we flip them.

In the initialization, with high probability at least $2^{6n-1}\log R_0$ of the points $\boldsymbol{x}_i$ will be chosen from $C_1 \cup C_2$, since by Lemma 4.4 we have $\Pr_{\boldsymbol{x}\sim\mathcal{B}(\mathbf{0},2)}[\boldsymbol{x} \in C_1] \geq 2^{-2n}$. We call those points in $C_1 \cup C_2$ *good*. During the algorithm, at most $O(\log R_0) \cdot 2^{5n}$ many points are removed (using that the algorithm takes only $O(\log R_0)$ iterations and in each one we remove at most 2^{5n} many cluster centers), hence when we arrive at step (6), there will be still more than, say 2^{5n} pairs left for which we have not decided the side. All the vectors $\boldsymbol{z}_i$ at that point (before tossing) will have $\|\boldsymbol{z}_i\|_2 \leq \|\boldsymbol{y}_i\|_2 + \|\boldsymbol{x}_i\|_2 \leq 6+2 = 8$. But by Lemma 4.5,

we have $|\mathcal{B}(\mathbf{0},8)\cap\Lambda(\boldsymbol{B})|\le 2^{4n}$. Hence there must be one lattice point $\boldsymbol{w}\in\mathcal{B}(\mathbf{0},8)\cap\Lambda(\boldsymbol{B})$ so that $\boldsymbol{z}_i=\boldsymbol{w}$ for at least $\frac{2^{5n}}{2^{4n}}\ge 2^{4n}$ points. With exponentially high probability, during tossing many of the vectors $\boldsymbol{z}_i$ will stay $\boldsymbol{w}$ — many others will be flipped to $\boldsymbol{w}+\boldsymbol{v}$ or $\boldsymbol{w}-\boldsymbol{v}$. Hence, if we consider the differences in step (6), then we will almost certainly find the shortest vector $\boldsymbol{v}$. That concludes the proof! □

4.4 Exercises

Exercise 4.1. Let $\Lambda\subseteq\mathbb{R}^n$ be a lattice and let λ be the length of the Shortest Vector. Show that $|\Lambda\cap\mathcal{B}(\mathbf{0},k\cdot\lambda)|\le(2k+1)^n$ for any $k\ge 1$.

Exercise 4.2. The algorithm can be easily modified to work with different norms. Describe on $\frac{1}{2}-1$ page how and why the algorithm also works for the $\|\cdot\|_\infty$-norm.

Exercise 4.3. Let $\Lambda\subseteq\mathbb{R}^n$ be a lattice with $2\le\text{SVP}(\Lambda)\le 3$. Describe a small modification of the algorithm that with high probability outputs all vectors in $\Lambda\cap\mathcal{B}(\mathbf{0},3)$.

Exercise 4.4. Let $\Lambda(\boldsymbol{B})$ be a lattice and let $\boldsymbol{B}$ be an LLL-reduced basis. Let $\boldsymbol{x}^*\in\Lambda(\boldsymbol{B})\setminus\{\mathbf{0}\}$ be the shortest lattice vector and let $\boldsymbol{\lambda}\in\mathbb{Z}^n$ be the vector with $\boldsymbol{x}^*=\boldsymbol{B}\boldsymbol{\lambda}$. Prove that $\|\boldsymbol{\lambda}\|_\infty\le 2^{1.5n}$.
Remark. This insight gives a simple algorithm that finds the shortest vector in time $2^{O(n^2)}$ by testing all points $\boldsymbol{B}\boldsymbol{\lambda}$ where $\boldsymbol{\lambda}\in\mathbb{Z}^n$ and $\|\boldsymbol{\lambda}\|_\infty\le 2^{1.5n}$.

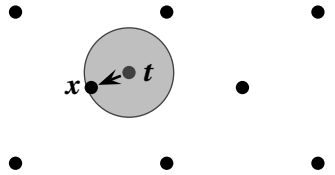
Chapter 5

The Closest Vector Problem

In this chapter, we will study a new problem:

> CLOSEST VECTOR PROBLEM (CVP)
> **Input:** A lattice $\Lambda(\boldsymbol{B})$ given by a regular matrix $\boldsymbol{B} \in \mathbb{Q}^{n\times n}$ and a target vector $\boldsymbol{t} \in \mathbb{R}^n$.
> **Goal:** Find the lattice vector $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ minimizing $\|\boldsymbol{x}-\boldsymbol{t}\|_2$.

A small example is as follows:

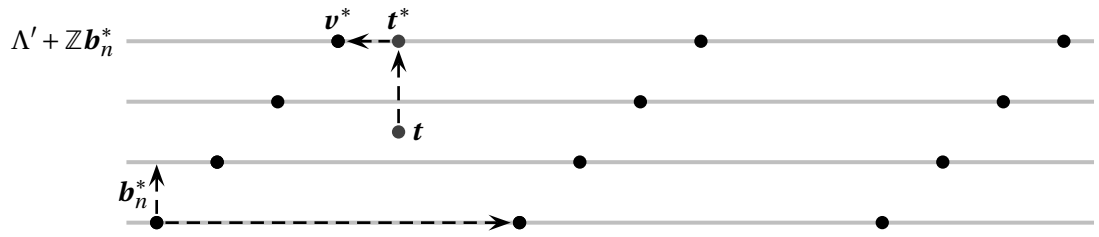


We will use the notation $\text{CVP}(\Lambda(\boldsymbol{B}), \boldsymbol{t})$ to denote the value of the optimum solution. One might imagine CVP as a more general version of the Shortest Vector problem (in the exercises, we will justify this claim). The main content of this chapter will be the ingenious algorithm by Micciancio[1] and Voulgaris [MV10, MV13].

5.1 A $2^{O(n^2)}$-algorithm for Closest Vector

To warm up, we want to describe a simple algorithm that solves CVP in time $2^{O(n^2)}$. So, let $\boldsymbol{B}$ be the given lattice basis. In a first step, we replace $\boldsymbol{B}$ with an *LLL-reduced basis*, which only takes polynomial time, see Chapter 1. Suppose that $(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*)$ is the *Gram-Schmidt orthogonalization*. That sequence of vectors has many properties, for example $\text{span}\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i\} = \text{span}\{\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_i^*\}$ for all i. Moreover, as $\boldsymbol{B}$ is LLL-reduced we know that $\|\boldsymbol{b}_1\|_2^2 = \|\boldsymbol{b}_1^*\|_2^2 \leq 2\cdot\|\boldsymbol{b}_2^*\|_2^2 \leq 4\|\boldsymbol{b}_3^*\|_2^2 \leq \ldots \leq 2^{n-1}\|\boldsymbol{b}_n^*\|_2^2$. Now, consider the target vector $\boldsymbol{t}$ and let $\Lambda' = \{\sum_{i=1}^{n-1}\lambda_i\boldsymbol{b}_i : \lambda_i \in \mathbb{Z}\}$ be the $(n-1)$-dimensional sublattice formed by the first $n-1$ vectors. By the properties of the Gram-Schmidt orthogonalization, the whole lattice can be covered by affine subspaces of the form $\text{span}(\Lambda') + k\boldsymbol{b}_n^*$ for $k \in \mathbb{Z}$. Let us call each such subspace a *layer*. If we want to find the closest lattice point $\boldsymbol{v}^* \in \Lambda$ to $\boldsymbol{t}$, then this can be done as follows: guess the right layer containing $\boldsymbol{v}^*$ and compute the orthogonal projection $\boldsymbol{t}^*$ of $\boldsymbol{t}$ on that layer. Then compute the closest vector to $\boldsymbol{t}^*$ with respect to the lower-dimensional lattice Λ'.

[1] The following link contains slides of Micciancio on the algorithm: `https://cseweb.ucsd.edu/~daniele/papers/Voronoi-slides.pdf`



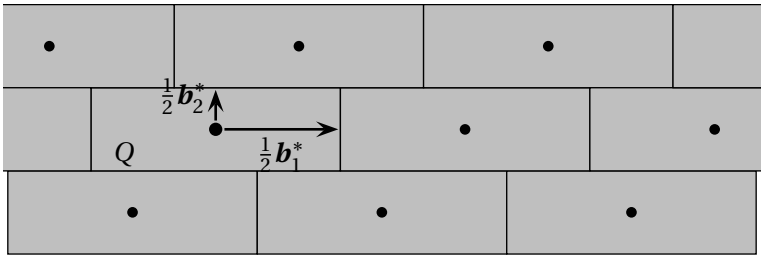
To make this recursive computation at least mildly efficient, we need to be able to bound the number of layers that we need to try out. In particular, $\boldsymbol{v}^*$ does not need to lie on the closest layer (as we can see in the picture above). But here is the crucial insight (still in the notation from above):

Lemma 5.1. *Let $\boldsymbol{t} \in \mathbb{R}^n$ be any target vector. Then the closest lattice point $\boldsymbol{v}^*$ lies on one of the at 2^n layers that are closest to $\boldsymbol{t}$.*

This claim follows immediately form the following:

Lemma 5.2. *Let $\Lambda(\boldsymbol{B})$ be any lattice with LLL-reduced basis $\boldsymbol{B}$ and Gram-Schmidt orthogonalization $\boldsymbol{b}_1^*, \dots, \boldsymbol{b}_n^*$. For any target vector $\boldsymbol{t}$, one has $\text{CVP}(\Lambda(\boldsymbol{B}), \boldsymbol{t}) \leq 2^n \|\boldsymbol{b}_n^*\|_2$.*

Proof. Let $Q := \{\sum_{i=1}^n \lambda_i \boldsymbol{b}_i^* : |\lambda_i| \leq \frac{1}{2}\}$ be a parallelepiped with center $\mathbf{0}$ whose sides are spanned by the Gram-Schmidt orthogonalization. Geometrically speaking, Q is a shifted and translated version of the fundamental parallelepiped of the basis $\boldsymbol{B}$. In particular, Q and $\mathcal{P}(\boldsymbol{B})$ have the same volume and both have the property that translates of them by lattice vectors exactly partition $\mathbb{R}^n$ (apart from a zero set).



Hence for each point $\boldsymbol{t} \in \mathbb{R}^n$, there is a lattice point $\boldsymbol{v} \in \Lambda(\boldsymbol{B})$ so that $\boldsymbol{t} \in \boldsymbol{v} + Q$. This point satisfies $\|\boldsymbol{t} - \boldsymbol{v}\|_2^2 \leq \sum_{i=1}^n \|\frac{1}{2}\boldsymbol{b}_i^*\|_2^2 \leq \sum_{i=1}^n 2^{n-i}\|\boldsymbol{b}_n^*\|_2^2 \leq 2^n \|\boldsymbol{b}_n^*\|_2^2$. □

Theorem 5.3. *The Closest vector problem can be solved in time $2^{O(n^2)}$.*

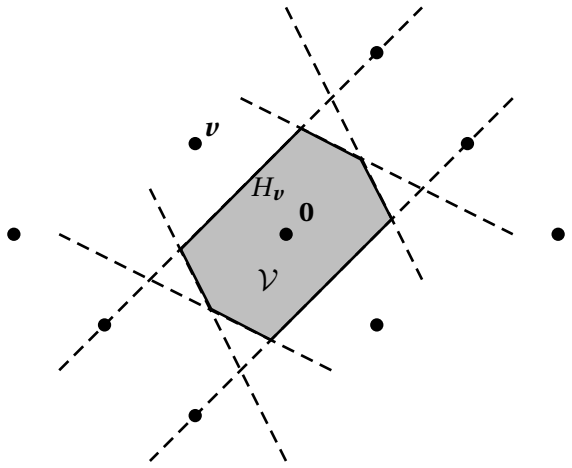
Proof. Let $T(n)$ be the running time to solve CVP in dimension n. Then we just discussed that $T(n) \leq 2^n \cdot T(n-1)$. Applying induction then gives the claim. □

Obviously this is just a simple and naive algorithm. The induction can be done in a smarter way, see the $n^{O(n)} \cdot \text{poly(input)}$ algorithm by Kannan [Kan87a]. But one might already see one potential for improvement: in each of the 2^n recursions we are solving the closest vector problem for the *same* lattice — just each time with different target vectors. If we could come up with some kind of preprocessing for the lattice, then we might reuse those computations in each of the recursions and speed up the algorithm.

5.2 The Voronoi cell

In this section, we want to explain an important concept that is the base for the algorithm of Micciancio and Voulgaris [MV10, MV13]. For a lattice $\Lambda(\boldsymbol{B})$, the *(open) Voronoi cell* is the set of points in $\mathbb{R}^n$ that is closer to $\mathbf{0}$ than to any other lattice point. Formally, we define

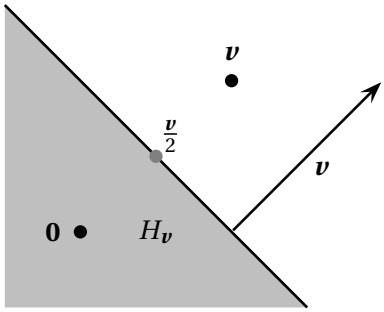
$$\mathcal{V} = \left\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 < \|\boldsymbol{x} - \boldsymbol{v}\|_2 \ \forall \boldsymbol{v} \in \Lambda(\boldsymbol{B}) / \{\mathbf{0}\}\right\}$$



Note that the set of points that are closer to $\mathbf{0}$ than to $\boldsymbol{v} \in \Lambda$ is exactly the set of points in the open halfspace

$$H_{\boldsymbol{v}} = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\|_2 < \|\boldsymbol{x} - \boldsymbol{v}\|_2\right\} = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \langle \boldsymbol{x}, \boldsymbol{v} \rangle < \frac{1}{2}\|\boldsymbol{v}\|_2^2\right\}$$

Geometrically, the normal vector of this halfspace is $\boldsymbol{v}$ and it contains the point $\frac{\boldsymbol{v}}{2}$ on its boundary:



Using this definition, we can write $\mathcal{V}$ as intersection of all those halfspaces

$$\mathcal{V} = \bigcap_{\boldsymbol{v} \in \Lambda(\boldsymbol{B}) / \{\mathbf{0}\}} H_{\boldsymbol{v}}$$

From that definition, we see that $\mathcal{V}$ is a symmetric, convex set. The set must be *full-dimensional* as it contains the open ball of radius $\frac{1}{2}\textsc{Svp}(\Lambda(\boldsymbol{B}))$ around the origin. Next, consider the set $Q := \bigcap_{\boldsymbol{v} \in \{\pm \boldsymbol{b}_1, \ldots, \pm \boldsymbol{b}_n\}} H_{\boldsymbol{v}}$. Since $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ are a basis, the set Q will be bounded. Then there are only finitely many lattice vectors inside of $2Q$ and any lattice vector outside of $2Q$ cannot induce a halfspace $H_{\boldsymbol{v}}$ cutting off even parts of Q. Hence $\mathcal{V}$ is described by *finitely* many half-spaces and it is an *(open) polytope*.

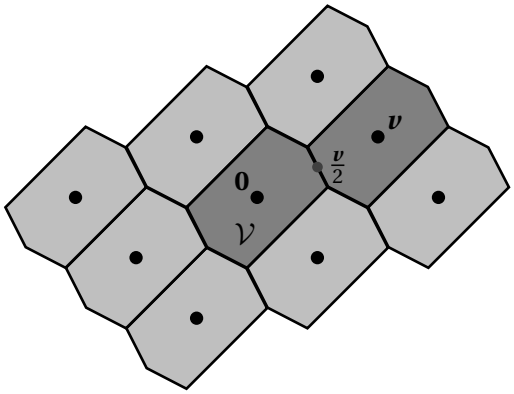
We also denote $\bar{\mathcal{V}}$ as the *closure* of the Voronoi cell. In other words, $\bar{\mathcal{V}} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x} - \boldsymbol{v}\|_2 \ \forall \boldsymbol{v} \in \Lambda(\boldsymbol{B}) / \{\mathbf{0}\}\}$. Every irredundant halfspace $\bar{H}_{\boldsymbol{v}}$ describing the cell must induce a $(n-1)$-dimensional facet of $\bar{\mathcal{V}}$. Hence there is a unique minimal set $R \subseteq \Lambda(\boldsymbol{B}) \setminus \{\mathbf{0}\}$ so that $\mathcal{V} = \bigcap_{\boldsymbol{v} \in R} H_{\boldsymbol{v}}$. We call that set R the *Voronoi-relevant vectors*.

Lemma 5.4. *If $\mathcal{V} \cap (\boldsymbol{v} + \mathcal{V}) \neq \emptyset$, then $\frac{\boldsymbol{v}}{2} \in \mathcal{V} \cap (\boldsymbol{v} + \mathcal{V})$.*

Proof. Suppose that $\frac{\boldsymbol{v}}{2} \notin \mathcal{V}$. Since $\mathcal{V}$ is convex, there is a hyperplane with $\langle \boldsymbol{a}, \boldsymbol{x} \rangle < \langle \boldsymbol{a}, \frac{\boldsymbol{v}}{2} \rangle$ for all $\boldsymbol{x} \in \mathcal{V}$. Since $\mathcal{V}$ is symmetric, that same hyperplane also has $\langle \boldsymbol{a}, \boldsymbol{x} \rangle > \langle \boldsymbol{a}, \frac{\boldsymbol{v}}{2} \rangle$ for all $\boldsymbol{x} \in \boldsymbol{v} + \mathcal{V}$. Then we have a hyperplane separating $\mathcal{V}$ and $\boldsymbol{v} + \mathcal{V}$. This is a contradicion. □

In particular this implies that for any $\boldsymbol{v} \in R$ one has that $\frac{\boldsymbol{v}}{2}$ lies in the interior of the facet induced by $H_{\boldsymbol{v}}$.

Another useful geometric observation is that if we place translates of the cell $\mathcal{V}$ at each lattice point, then we can exactly partition the space $\mathbb{R}^n$ (except for the zero-set of points that has equal minimum distance to two lattice points).



Now, things are getting a little more interesting as we show that there is a limited number of Voronoi relevant vectors:

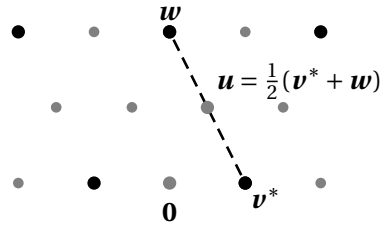
Lemma 5.5. *The number of Voronoi relevant vectors is $|R| \leq 2^{n+1}$.*

Proof. Let $X := \{\sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i \mid \lambda_i \in \{0,1\}\}$ be the vertices of the fundamental parallelepiped of the basis $\boldsymbol{B}$. Note that $|X| = 2^n$. Fix a vector $\boldsymbol{v} \in X$ and consider the translated lattice $\boldsymbol{v} + 2\Lambda$. We claim that *at most one* of the lattice vectors in that translated lattice was Voronoi relevant. In fact, we claim something stronger:
Claim: Apart from $\boldsymbol{v}^* := \operatorname{argmin}\{\|\boldsymbol{x}\|_2 : \boldsymbol{x} \in \boldsymbol{v} + 2\Lambda\}$ and $-\boldsymbol{v}^*$, there is no other Voronoi relevant vector in $\boldsymbol{v} + 2\Lambda$.
Proof: Suppose that $\boldsymbol{w} \in \Lambda(\boldsymbol{B})$ is Voronoi-relevant. Then $\frac{\boldsymbol{w}}{2}$ lies in the interior of an $(n-1)$-dimensional facet of $\bar{\mathcal{V}}$ and the set of lattice vectors minimizing the distance to $\frac{\boldsymbol{w}}{2}$ is precisely $\{\boldsymbol{0}, \boldsymbol{w}\}$. We will bring this to a contradiction by showing that $\frac{\boldsymbol{w}}{2}$ lies outside of $H_{\boldsymbol{u}}$ for another lattice vector u.

Let $\boldsymbol{v}^*$ be the shortest lattice vector of the same parity as $\boldsymbol{w}$, that means $\boldsymbol{v}^* = \text{SVP}(\boldsymbol{w} + 2\Lambda)$. We know that $\boldsymbol{u} := \frac{1}{2}(\boldsymbol{v}^* + \boldsymbol{w}) \in \Lambda(\boldsymbol{B})$. We claim that the distance of the midpoint $\frac{\boldsymbol{w}}{2}$ to $\boldsymbol{u}$ is not larger than the distance to $\boldsymbol{0}$ and $\boldsymbol{w}$, which implies that $\boldsymbol{w}$ was not Voronoi relevant. To see this, calculate

$$\left\| \frac{\boldsymbol{w}}{2} - \boldsymbol{u} \right\|_2 = \left\| \frac{\boldsymbol{w}}{2} - \frac{1}{2}(\boldsymbol{v}^* + \boldsymbol{w}) \right\|_2 = \frac{1}{2}\|\boldsymbol{v}^*\|_2 \leq \frac{1}{2}\|\boldsymbol{w}\|_2$$

The claim follows from the contradiction.



□

The last lemma also provides us with a possibility to compute the Voronoi relevant vectors:

Lemma 5.6. *The Voronoi relevant vectors for a Voronoi cell $\mathcal{V}$ for lattice $\Lambda(\boldsymbol{B})$ can be computed by $2^{O(n)}$ many* CVP *calls with the same lattice $\Lambda(\boldsymbol{B})$ (but different target vectors).*

Proof. For every $\boldsymbol{\lambda} \in \{0,1\}^n$ we need to find the shortest vector in the shifted lattice $\boldsymbol{t} + 2\Lambda$ with $\boldsymbol{t} := \sum_{i=1}^n \lambda_i \boldsymbol{b}_i$. That is the same as solving $\text{CVP}(2\Lambda(\boldsymbol{B}), -\boldsymbol{t})$ which is the same as $\text{CVP}(\Lambda(\boldsymbol{B}), -\frac{\boldsymbol{t}}{2})$. □

It seems that if we want to use the Voronoi cell in an algorithm for CVP, we first have to solve $2^{O(n)}$ instances of CVP. This seems utterly stupid — but it will work!

5.3 Computing a closest vector

We can now come to the main algorithm which on input $\boldsymbol{t} \in \mathbb{R}^n$ computes a closest lattice vector $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$. Here, we assume that we do have a description of the Voronoi cell in form of the Voronoi relevant vectors R. Note that this is task is equivalent to finding a lattice vector $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ so that $\boldsymbol{t} - \boldsymbol{x} \in \mathcal{V}$. The algorithm is now as follows: starting at $\boldsymbol{t}$, subtract iteratively multiples of lattice vectors in R until we reach a point in the cell $\mathcal{V}$. Then the sum of the subtracted vectors will be the optimum solution $\boldsymbol{x}$. A detailed description is as follows:

Closest vector algorithm

Input: A lattice basis $\boldsymbol{B} \in \mathbb{R}^{n \times n}$, a target vector $\boldsymbol{t} \in \mathbb{R}^n$, the list of Voronoi relevant vectors $R \subseteq \Lambda(\boldsymbol{B}) / \{\boldsymbol{0}\}$ describing the Voronoi cell $\mathcal{V}$

Output: The lattice vector $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ minimizing $\|\boldsymbol{t} - \boldsymbol{x}\|_2$

(1) Set $\boldsymbol{s} := \boldsymbol{t}$

(2) WHILE $\boldsymbol{s} \notin \mathcal{V}$ DO

  (3) Compute the minimal value $\delta > 0$ so that $\boldsymbol{s} \in \delta \cdot \bar{\mathcal{V}}$

  (4) Write $\delta = 2^k \cdot \alpha$ with $k \in \mathbb{Z}_{\geq 0}$ and $1 \leq \alpha < 2$

  (5) Find the vector $\boldsymbol{v} \in R$ so that $\boldsymbol{s}$ lies on the boundary of $\delta H_{\boldsymbol{v}}$

  (6) Update $\boldsymbol{s} := \boldsymbol{s} - 2^k \boldsymbol{v}$

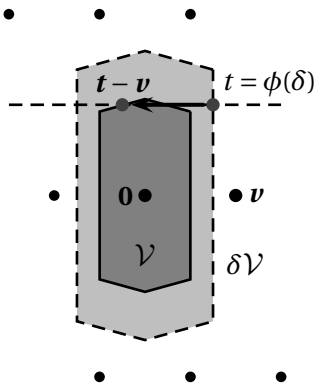
(7) Return $\boldsymbol{t} - \boldsymbol{s}$

For the analysis, it suffices to consider the situation with $k = 0$. In other words, we assume that the original vector $\boldsymbol{t}$ satisfies $\boldsymbol{t} \in 2\mathcal{V}$. We will show that within $2^{O(n)}$ iterations, we reach a point in $\mathcal{V}$. The general claim follows from the fact that $2^k \mathcal{V}$ is exactly the Voronoi cell of the scaled lattice $\Lambda(2^k \boldsymbol{B})$. Hence the algorithm iterates only over polynomially many choices of k.

So, let us understand what happens in one iteration.

Lemma 5.7. *Let $\boldsymbol{t} \in (2\mathcal{V}) / \mathcal{V}$ and let $1 \leq \delta < 2$ be the minimal value so that $\boldsymbol{t} \in \delta \bar{\mathcal{V}}$. Let $\boldsymbol{v} \in R$ be the Voronoi relevant vector so that $\boldsymbol{t}$ lies on the boundary of $\delta H_{\boldsymbol{v}}$. Then $\boldsymbol{t} - \boldsymbol{v} \in 2\mathcal{V}$ and $\|\boldsymbol{t} - \boldsymbol{v}\|_2 < \|\boldsymbol{t}\|_2$.*

Proof. Let $\phi : \mathbb{R} \to \mathbb{R}^n$ be the line with direction vector $\boldsymbol{v}$ through $\boldsymbol{t}$ parameterized so that $\phi(\delta) = \boldsymbol{t}$, and $\phi(\delta + 1) = \boldsymbol{t} + \frac{\boldsymbol{v}}{2}$. Then $\|\phi(\lambda)\|_2$ is a symmetric, convex function which is minimized for $\lambda = 0$. Hence $\|\boldsymbol{t} - \boldsymbol{v}\|_2 = \|\phi(\delta - 2)\|_2 < \|\phi(\delta)\|_2$ since $|\delta - 2| < |\delta|$.



From the argument we also see that $\boldsymbol{t}-\boldsymbol{v} \in \delta\mathcal{V}$ and by convexity and symmetry of V, also $\boldsymbol{t}-\boldsymbol{v} \in 2\mathcal{V}$. □

The algorithm computes shorter and shorter vectors in $2\mathcal{V}$, so it will definitely terminate in finite time. It remains to show that only $2^{O(n)}$ iterations are needed.

Lemma 5.8. *Let $\mathcal{V}$ be the Voronoi cell of a lattice $\Lambda \subseteq \mathbb{R}^n$ and let $\boldsymbol{t} \in 2\mathcal{V}$. Then $|(\boldsymbol{t}-\Lambda) \cap 2\mathcal{V}| \leq 2^{O(n)}$.*

Proof. Since $\boldsymbol{t} \in 2\mathcal{V}$, it suffices to show that $|\Lambda \cap 4\mathcal{V}| \leq 2^{O(n)}$. Suppose $|\Lambda \cap 4\mathcal{V}| > 4^n$. By a counting argument, there must be two distinct vectors $\boldsymbol{x}, \boldsymbol{y} \in 4\mathcal{V} \cap \Lambda$ with so that $\boldsymbol{x}-\boldsymbol{y} \in 4\Lambda \setminus \{\boldsymbol{0}\}$. Then $\boldsymbol{v} := \frac{1}{4}(\boldsymbol{x}-\boldsymbol{y})$ has $\boldsymbol{v} \in \Lambda$ and $\boldsymbol{v} \in \mathcal{V}$. But there is no lattice vector in $\mathcal{V}$ except $\boldsymbol{0}$. □

The lemma shows that for every fixed value of k, the algorithm iteratively finds shorter vectors and hence will not revisit a vector. Then after at most $2^{O(n)}$ iterations, the value of k will be decreased by one.

5.4 Putting things together

Setting up the recursion is not completely trivial in this case. Let us define two running times that we want to analyze

$$\begin{aligned} T_{\text{Voronoi}}(n) &= \text{time to compute the Voronoi cell in an } n\text{-dim. lattice} \\ T_{\text{CVP}}(n,k) &= \text{time to solve } k \text{ many CVP in the same } n\text{-dim. lattice} \end{aligned}$$

We can get the following sequence of recursions:

$$T_{\text{Voronoi}}(n) \overset{(1)}{\leq} T_{\text{CVP}}(n, 2^{O(n)}) \overset{(2)}{\leq} T_{\text{CVP}}(n-1, 2^{O(n)} \cdot 2^{O(n)}) \overset{(3)}{\leq} T_{\text{Voronoi}}(n-1) + 2^{O(n)} \cdot 2^{O(n)} \cdot 2^{O(n)}.$$

Here we use in (1) that we can compute the Voronoi cell with $2^{O(n)}$ CVP computations in the same lattice. In (2), we use the dimension reduction argument from Section 5.1 saying that a CVP computation can be reduced to $2^{O(n)}$ CVP computations in the same $(n-1)$-dimensional lattice. Finally, in (3) we use that to solve k many CVP computations in the same lattice, we need to compute the Voronoi cell only *once* and then run a $2^{O(n)}$-time algorithm for each of the k target vectors. From the recursion we conclude that $T_{\text{Voronoi}}(n) \leq \sum_{k=1}^{n} 2^{O(k)} = 2^{O(n)}$ and $T_{\text{CVP}}(n,1) \leq 2^{O(n)}$ as well.

5.5 Exercises

Exercise 5.1. One might wonder whether the algorithm can be modified to work with different norms, say with $\|\cdot\|_\infty$. To examine this, consider the points $(0,0)$ and $(2,1)$ in $\mathbb{R}^2$ and draw the region of points that is closer to $(0,0)$ than to $(2,1)$ with respect to the $\|\cdot\|_\infty$-norm. What do you think, can you guarantee that Voronoi cells with respect to $\|\cdot\|_\infty$ are convex?

Exercise 5.2. In this exercise, we want to show that SVP is not harder than CVP in the sense that we can use an oracle for CVP to solve the SVP problem. We denote $\text{CVP}(\boldsymbol{B}', \boldsymbol{t}) := \text{argmin}\{\|\boldsymbol{x} - \boldsymbol{t}\|_2 : \boldsymbol{x} \in \Lambda(\boldsymbol{B}')\}$. Suppose that $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is the input basis for our SVP problem. Consider the following algorithm:

(1) FOR $i = 1$ TO n DO

  (2) Set $\boldsymbol{v}_i := \text{CVP}((\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}, 2\boldsymbol{b}_i, \boldsymbol{b}_{i+1}, \ldots, \boldsymbol{b}_n), \boldsymbol{b}_i)$.

(3) Return the shortest vector in $\{\boldsymbol{v}_i - \boldsymbol{b}_i \mid i = 1, \ldots, n\}$

Note that the algorithm calls the CVP oracle only n times on a lattice of dimension n. Prove that the algorithm returns the shortest vector in $\Lambda(\boldsymbol{B})$.
Remark: There is no natural reduction known that goes the other way. Both problems are **NP**-hard, so there will be *some* reduction. But any known reduction from CVP to SVP causes at least a quadratic blowup in the dimension.

Exercise 5.3. Let $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ be a regular matrix. Prove that $\mathcal{V} \subseteq \mathcal{B}(\boldsymbol{0}, n \cdot \lambda_n(\Lambda))$ where $\mathcal{V}$ is the Voronoi cell of the lattice $\Lambda := \Lambda(\boldsymbol{B})$.

Exercise 5.4. For a lattice Λ let us write $\text{SVP}(\Lambda)$ and $\text{CVP}(\Lambda, t)$ as the values of the shortest vector and closest vector problems. We have seen in an earlier exercise that for any lattice Λ, the number of lattice vectors of length, say $2 \cdot \text{SVP}(\Lambda)$ is bounded by $2^{O(n)}$. Here we want to show that this is not true anymore for CVP. To be precise, for any function $f(n)$, construct a lattice in n dimensions and a point $\boldsymbol{t}$ so that $|\{\boldsymbol{x} \in \Lambda \mid \|\boldsymbol{x} - \boldsymbol{t}\|_2 \leq 2 \cdot \text{CVP}(\Lambda, \boldsymbol{t})\}| \geq f(n)$.

Exercise 5.5. Let $f(n) \leq O(n^{4/3} \cdot \text{polylog}(n))$ be the flatness constant. That means for each convex body K with $K \cap \mathbb{Z}^n = \emptyset$, there is a direction $\boldsymbol{c} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$ with $w_{\boldsymbol{c}}(K) \leq f(n)$. Use this (non-constructive!) result and the CVP algorithm to give a $n^{O(n)}$ times poly-time algorithm for integer programming.

Exercise 5.6. Give a formal proof for the following claim: For any lattice basis $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ with Gram-Schmidt orthogonalization $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ define $Q := \{\sum_{i=1}^n \lambda_i \boldsymbol{b}_i^* \mid \frac{1}{2} \leq \lambda_i < \frac{1}{2}\ \forall i \in [n]\}$. Then $\boldsymbol{x} + Q$ for $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ form a *tiling* of $\mathbb{R}^n$ (meaning that for each $\boldsymbol{t} \in \mathbb{R}^n$ there is exactly one $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ with $\boldsymbol{t} \in \boldsymbol{x} + Q$).

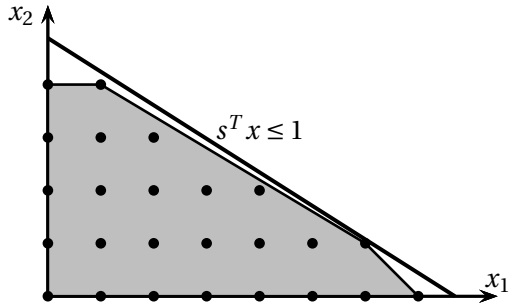
Chapter 6

Integer conic combinations

Imagine you are a manufacturer of pipes; your production spits out large quantities of pipes that all have a length of exactly 1 meter and for technical reasons that process cannot be modified. Further imagine that a customer gives you a large order where he orders $a_i \in \mathbb{Z}_{\geq 0}$ many pipes of length $s_i \in [0,1]$. Let d be the number of different sizes that your customer orders. We, as a manufacturer can now cut our 1-meter pipes into smaller pipes in order to satisfy our customer demand. The abstract version of this problem is called the *cutting stock problem*.

CUTTING STOCK
Input: A size vector $\boldsymbol{s} \in [0,1]^d$ and a multiplicity vector $\boldsymbol{a} \in \mathbb{Z}^d_{\geq 1}$.
Goal: Find the minimum number of vectors from $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{Z}^d_{\geq 0} \mid \sum_{i=1}^d s_i x_i \leq 1\}$ (with repetition) whose sum gives $\boldsymbol{a}$.

The set $\mathcal{P}$ is also called the set of *patterns*. In our example application, each vector in $\mathcal{P}$ corresponds to one possible option how a single 1-meter pipe can be cut into smaller pipes. A different way to interpret the patterns is that $\mathcal{P}$ is the set of integer points in the *Knapsack polytope* $P = \text{conv}\{\boldsymbol{x} \in \mathbb{R}^d_{\geq 0} \mid \sum_{i=1}^d s_i x_i \leq 1\}$.



Knapsack polytope for $s = (0.13, 0.205)$.

The attentative reader might already have realized that already for $a_i = 1$, this problem is the well-known *bin packing* problem. Hence, the problem is at least **NP**-hard and there is no hope for a polynomial time algorithm in general. In this chapter we are particularly interested in the case that the multiplicities a_i are very large. There are some non-trivial questions that we will try to answer:

- *Question 1*: Is the CUTTING STOCK problem in **NP**? This is not clear because the output might be exponential in the input size. It turns out that the solution is implied by answering the next question affirmatively.
- *Question 2*: Is there always an optimum solution where only a polynomial number of different vectors from $\mathcal{P}$ are chosen? Here, polynomial means the number is of the form $\text{poly}(d, \log \|\boldsymbol{a}\|_\infty, \max_i \log(\frac{1}{s_i}))$.
- *Question 3*: If d is a constant, is there a polynomial time algorithm? Again, the problem is that even for $d = 2$ the number of patterns in $\mathcal{P}$ as well as the output size may be exponentially large.

6.1 Small support for integer conic combinations

We will start the discussion by giving a bound on the number of different patterns that are needed. The arguments nicely extend to a more general setting. Suppose that $X \subseteq \mathbb{R}^d$ is some set of points. Then

$$\text{int.cone}(X) := \Big\{ \sum_{\boldsymbol{x} \in X} \lambda_{\boldsymbol{x}} \cdot \boldsymbol{x} \mid \lambda_{\boldsymbol{x}} \in \mathbb{Z}_{\geq 0} \ \forall \boldsymbol{x} \in X \Big\}$$

is the set of *integer conic combinations*. In other words, $\text{int.cone}(X)$ is the set of all vectors that can be obtained by adding up vectors in X — where arbitrary repetition is allowed. The reader is probably familiar with the "fractional" version

$$\text{cone}(X) := \Big\{ \sum_{\boldsymbol{x} \in X} \lambda_{\boldsymbol{x}} \cdot \boldsymbol{x} \mid \lambda_{\boldsymbol{x}} \in \mathbb{R}_{\geq 0} \ \forall \boldsymbol{x} \in X \Big\}$$

$\text{int.cone}(X)$ $\text{cone}(X)$

The classical *Theorem of Carathéodory* says that if $\boldsymbol{b} \in \text{cone}(X)$, then there is a conic combination $\boldsymbol{\lambda} \in \mathbb{R}^X_{\geq 0}$ with only $|\text{supp}(\boldsymbol{\lambda})| \leq d$ many non-zero entries and $\boldsymbol{b} = \sum_{\boldsymbol{x} \in X} \lambda_{\boldsymbol{x}} \cdot \boldsymbol{x}$. It turns out that there are extensions to integer conic combinations — but the situation is more complicated:

Lemma 6.1 (Eisenbrand and Shmonin [ES06])**.** *Let $X \subseteq \mathbb{Z}^d_{\geq 0}$ and $\boldsymbol{b} \in \text{int.cone}(X)$. Then there is a subset $Y \subseteq X$ of size $|Y| \leq d \cdot \log(\|\boldsymbol{b}\|_\infty + 1)$ with $\boldsymbol{b} \in \text{int.cone}(Y)$.*

Proof. We assume for the sake of contradiction that X with $|X| > d \cdot \log_2(\|\boldsymbol{b}\|_\infty + 1)$ is minimal in the sense that for all $Y \subset X$ one has $\boldsymbol{b} \notin \text{int.cone}(Y)$. We write $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|X|}\}$ and let $\boldsymbol{\lambda} \in \mathbb{Z}^{|X|}_{\geq 1}$ be the integer conic combination with $\boldsymbol{b} = \sum_{i=1}^{|X|} \lambda_i \boldsymbol{x}_i$. Consider the function

$$f : \{0,1\}^{|X|} \to \mathbb{Z}^d_{\geq 0} \quad \text{with} \quad f(\boldsymbol{a}) := \sum_{i=1}^{|X|} a_i \boldsymbol{x}_i.$$

Obviously the pre-image has size $2^{|X|}$. On the other hand, for any $\boldsymbol{a} \in \{0,1\}^{|X|}$ one has $\|f(\boldsymbol{a})\|_\infty \leq \|\sum_{i=1}^{|X|} \boldsymbol{x}_i\|_\infty \leq \|\sum_{i=1}^{|X|} \lambda_i \boldsymbol{x}_i\|_\infty = \|\boldsymbol{b}\|_\infty$ using that $\boldsymbol{x}_i \geq \boldsymbol{0}$ for all i. Hence the size of the image is at most

$$(1 + \|\boldsymbol{b}\|_\infty)^d = 2^{d \log(\|\boldsymbol{b}\|_\infty + 1)} < 2^{|X|}$$

By the *pigeonhole principle*, there is a collision in the form of vectors $\boldsymbol{a}, \boldsymbol{a}' \in \{0,1\}^{|X|}$ with $f(\boldsymbol{a}) = f(\boldsymbol{a}')$. We set $\boldsymbol{c} := \boldsymbol{a} - \boldsymbol{a}' \in \{-1,0,1\}^{|X|}$ with $f(\boldsymbol{c}) = \boldsymbol{0}$. Choose $s \in \mathbb{Z}_{\geq 0}$ as the largest integer so that $\lambda_i + s \cdot c_i \geq 0$ for all i. Then we still have

$$\boldsymbol{b} = \sum_{i=1}^{|X|} \underbrace{(\lambda_i + s \cdot c_i)}_{\in \mathbb{Z}_{\geq 0}} \cdot \boldsymbol{x}_i,$$

but for at least one i one has $\lambda_i + s \cdot c_i = 0$. That gives a smaller integer conic combination as assumed. □

The arguments generalize to the case where the vectors in X are not necessarily non-negative — we leave this case for the exercises. Note that the algorithm used the pigeonhole principle; hence even if a starting integer conic combination $\boldsymbol{\lambda}$ is given, it is not clear whether one could compute another one that is sparse. For CUTTING STOCK this implies the following:

Corollary 6.2. *The* CUTTING STOCK *problem is in* **NP**. *In particular for each instance* $(\boldsymbol{s}, \boldsymbol{a})$*, there is a solution using at most* $O(d \cdot \log \|\boldsymbol{a}\|_\infty)$ *many different patterns.*

6.2 Small support for integer conic cominations of convex sets

We will see in the exercises that the necessary support cannot be bounded by a function that depends only on d. But it turns out that under stronger assumptions, this is possible. The following result is again due to Eisenbrand and Shmonin [ES06]:

Lemma 6.3. *For any convex set* $P \subseteq \mathbb{R}^d$ *and* $\boldsymbol{b} \in$ *int.cone*$(P \cap \mathbb{Z}^d)$ *there is a subset* $X \subseteq P \cap \mathbb{Z}^d$ *of size* $|X| \leq 2^d$ *so that* $\boldsymbol{b} \in$ *int.cone*(X).

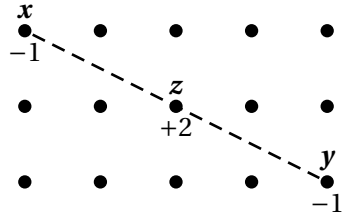
Proof. As $\boldsymbol{b} \in \text{int.cone}(P \cap \mathbb{Z}^d)$, there is a non-negative integral vector $(\lambda_{\boldsymbol{x}})_{\boldsymbol{x} \in P \cap \mathbb{Z}^d}$ so that $\boldsymbol{b} = \sum_{\boldsymbol{x} \in P \cap \mathbb{Z}^d} \lambda_{\boldsymbol{x}} \boldsymbol{x}$. For the sake of simplicity we can replace the original P with $P := \text{conv}\{\boldsymbol{x} \mid \lambda_{\boldsymbol{x}} > 0\}$ without changing the claim, where $\boldsymbol{\lambda}$ is the vector with $\boldsymbol{b} = \sum_{\boldsymbol{x}} \lambda_{\boldsymbol{x}} \boldsymbol{x}$. In particular this means that we can assume that P is bounded.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be any strictly convex function — for example $f(\boldsymbol{x}) = \|(1, \boldsymbol{x})\|_2$ does the job. We will only use that

$$f(\tfrac{1}{2}\boldsymbol{x} + \tfrac{1}{2}\boldsymbol{y}) < \frac{1}{2}(f(\boldsymbol{x}) + f(\boldsymbol{y})).$$

Note that the integer conic combination might not be unique — in that case, among all such candidate vectors $\boldsymbol{\lambda}$, we want to choose that vector that minimizes the potential function $\sum_{\boldsymbol{x} \in P \cap \mathbb{Z}^d} \lambda_{\boldsymbol{x}} \cdot f(\boldsymbol{x})$ (the minimum is attained for compactness reasons since $P \cap \mathbb{Z}^d$ is finite). In other words, we somewhat prefer points that are more in the "center" of the polytope. We claim that indeed $|\text{supp}(\boldsymbol{\lambda})| \leq 2^d$.

For the sake of contradiction suppose that $|\text{supp}(\boldsymbol{\lambda})| > 2^d$. Then there must be two points $\boldsymbol{x}, \boldsymbol{y}$ with $\lambda_{\boldsymbol{x}} > 0$ and $\lambda_{\boldsymbol{y}} > 0$ that have the same *parity*, meaning that $x_i \equiv y_i \mod 2$ for all $i = 1, \ldots, d$. Then $\boldsymbol{z} := \frac{1}{2}(\boldsymbol{x} + \boldsymbol{y})$ is an integral vector and $\boldsymbol{z} \in P$. Now we remove one unit of weight from both $\boldsymbol{x}$ and $\boldsymbol{y}$ and add 2 units to $\boldsymbol{z}$.



This gives us another feasible vector $\boldsymbol{\lambda}'$. But the change in the potential function is $+2f(\boldsymbol{z}) - f(\boldsymbol{x}) - f(\boldsymbol{y}) < 0$ by strict convexity of f, contradicting the minimality of $\boldsymbol{\lambda}$. □

In the exercises we will see that a support of 2^{d-1} might actually be needed.

6.3 An algorithm for Integer Conic Programming

In this section we want to discuss an algorithm that solves the CUTTING STOCK problem in polynomial time, if the number d is a fixed constant. In fact, we will see that even the more general problem can be solved in polynomial time:

INTEGER CONIC PROGRAMMING
Input: A polytope $P = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ and a target vector $\boldsymbol{a} \in \mathbb{Z}^d$.
Goal: Select points in $P \cap \mathbb{Z}^d$ (with repetition) whose sum gives $\boldsymbol{a}$.

Note that this is only a *decision version* and does not contain any objective function. But if we had a CUTTING STOCK instance $(\boldsymbol{s}, \boldsymbol{a})$ then we can check whether the objective function is at most k by testing whether INTEGER CONIC PROGRAMMING with the instance

$$P = \left\{ \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} \in \mathbb{R}^{d+1} \mid \sum_{i=1}^{d} s_i x_i \leq 1 \right\} \quad \text{and} \quad \boldsymbol{a}' = \begin{pmatrix} \boldsymbol{a} \\ k \end{pmatrix}$$

has a solution. Then we can use *binary search* to find the minimum k for which this is feasible which will then give the actual optimum value (and solution).

For the remainder of this chapter, we follow the paper by Goemans and Rothvoss [GR14]. Suppose that $\boldsymbol{a} \in \text{int.cone}(P \cap \mathbb{Z}^d)$ (otherwise there is nothing to show) and let us first discuss where the challenges lie:

- Lemma 6.3 tells us that there is an integer conic combination that uses at most 2^d many points. The problem is that $P \cap \mathbb{Z}^d$ might be exponentially large and we do not know *which* of the points are needed.

- If someone could tell us a set $X \subseteq P \cap \mathbb{Z}^d$ with $\boldsymbol{a} \in \text{int.cone}(X)$, then we can find the coefficients

  $$\boldsymbol{a} = \sum_{\boldsymbol{x} \in X} \lambda_{\boldsymbol{x}} \boldsymbol{x}$$

  using an integer linear program with a $|X|$ many variables. That problem can be solved via Lenstra's algorithm from Chapter 2.

- On the other hand, if someone told us that there is a solution with coefficients $\lambda_1, \ldots, \lambda_m$, then finding the corresponding points is again an integer linear program

  $$\exists (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \in \mathbb{Z}^{d \cdot m} : \sum_{i=1}^{m} \lambda_i \boldsymbol{x}_i = \boldsymbol{a} \text{ and } \boldsymbol{A}\boldsymbol{x}_i \leq \boldsymbol{b} \text{ for } i = 1, \ldots, m$$

  with $m \cdot d$ variables.

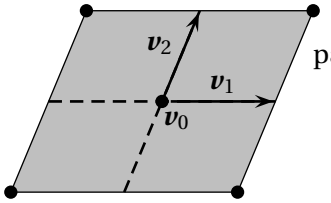
It will turn out that we can restrict the number of points that are useful for an integer linear combination.

6.3.1 A special case of parallelepipeds

We want to discuss how to solve INTEGER CONIC PROGRAMMING for a family of versatile and well-behaved polytopes. Recall that

$$\Pi = \left\{ \boldsymbol{v}_0 + \sum_{i=1}^{k} \mu_i \boldsymbol{v}_i \mid -1 \le \mu_i \le 1 \ \forall i = 1, \dots, k \right\}$$

is a *parallelepiped* with *center* $\boldsymbol{v}_0 \in \mathbb{R}^d$ and *directions* $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k \in \mathbb{R}^d$. Usually one requires that the directions are linearly independent, that means $k \le d$ and Π is k-dimensional. We say that the parallelepiped is *integral* if all its 2^k many vertices are integral.



parallelepiped Π

One might wonder whether it is true that any vector $\boldsymbol{a} \in \text{int.cone}(\Pi \cap \mathbb{Z}^n)$ can be written as integer conic combination of it's vertices. If true, that would solve our problem for parallelepipeds since Π has only 2^d (i.e. constant) many vertices. It turns out that this is *almost* true in the sense that only a constant number of copies from interior points are needed:

Lemma 6.4. *Given an integral parallelepiped Π with vertices $X := vert(\Pi)$. For any $\boldsymbol{a} \in int.cone(\Pi \cap \mathbb{Z}^n)$ there are $m \le 2^d$ points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in P \cap \mathbb{Z}^d$ so that $\boldsymbol{a} \in int.cone(X) + \sum_{i=1}^{m} \boldsymbol{x}_i$.*

Proof. Let $\Pi = \{\boldsymbol{v}_0 + \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i \mid |\alpha_i| \le 1 \ \forall i = 1, \dots, k\}$ where $\boldsymbol{v}_0$ is the (not necessarily integral) center of Π. By assumption, there is a vector $\boldsymbol{\lambda} \in \mathbb{Z}_{\ge 0}^{\Pi \cap \mathbb{Z}^d}$ so that

$$\boldsymbol{a} = \sum_{\boldsymbol{x} \in \Pi \cap \mathbb{Z}^d} \lambda_{\boldsymbol{x}} \cdot \boldsymbol{x} \qquad (*).$$

Among all possible choices for $\boldsymbol{\lambda}$ we select the one that minimizes the *potential function* $\sum_{\boldsymbol{x} \notin X} \lambda_{\boldsymbol{x}}$. In other words, we consider that integer conic combination of $\boldsymbol{a}$ that puts as little weight as possible on the non-vertices. Our claim is that $\boldsymbol{\lambda}$ satisfies the two conditions

$$(**) \quad \sum_{\boldsymbol{x} \notin X} \lambda_{\boldsymbol{x}} \le 2^d \qquad \text{and} \qquad (***) \quad \lambda_{\boldsymbol{x}} \in \{0, 1\} \quad \forall \boldsymbol{x} \notin X$$

First consider the case that there is some point $\boldsymbol{x}$ that is not a vertex and has $\lambda_{\boldsymbol{x}} \ge 2$. We write $\boldsymbol{x} = \boldsymbol{v}_0 + \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i$ with $|\alpha_i| \le 1$. Let[1] $\boldsymbol{y} := \boldsymbol{v}_0 + \sum_{i=1}^{k} \text{sign}(\alpha_i) \cdot \boldsymbol{v}_i$ be the vertex of Π that we obtain by rounding α_i to ± 1, see Figure 6.1. Note that the mirrored point $\boldsymbol{z} = \boldsymbol{x} + (\boldsymbol{x} - \boldsymbol{y}) = \boldsymbol{v}_0 + \sum_{i=1}^{k} (2\alpha_i - \text{sign}(\alpha_i)) \cdot \boldsymbol{v}_i$ lies in Π as well and is also integral. As $\boldsymbol{x} = \frac{1}{2}(\boldsymbol{y} + \boldsymbol{z})$, we can reduce the weight on $\boldsymbol{x}$ by 2 and add 1 to $\lambda_{\boldsymbol{y}}$ and $\lambda_{\boldsymbol{z}}$. We obtain again a vector that satisfies (1), but the weight $\sum_{\boldsymbol{x} \notin X} \lambda_{\boldsymbol{x}}$ has decreased.

So it remains to see what happens when all vectors in $(\Pi \cap \mathbb{Z}^d) \setminus X$ carry weight at most 1. Well, if these are at most 2^d, then we are done. Otherwise, we can reiterate the arguments from Lemma 6.3: there will be 2 points of the same parity, which can be joined to create a new point carrying weight at least 2 and part of this weight can be redistributed to a vertex. This shows the claim. □

[1] Recall that $\text{sign}(\alpha) = \begin{cases} 1 & \alpha \ge 0 \\ -1 & \alpha < 0 \end{cases}$

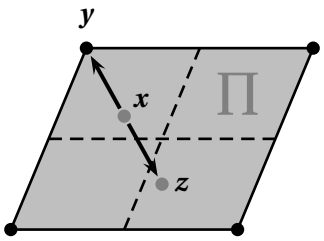


Figure 6.1: Weight of $\boldsymbol{y}$ is redistributed to vertex in parallelepiped.

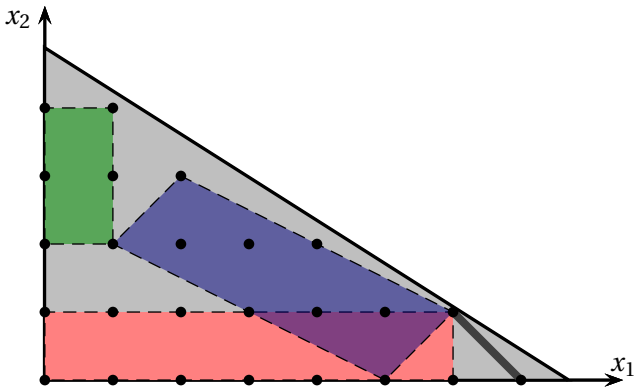


Figure 6.2: Covering the integer points of a polytope with integral parallelepipeds.

We are now able to solve the CONIC INTEGER PROGRAMMING problem for parallelepid Π: for a given target vector $\boldsymbol{a}$, guess[2] the number $m \in \{0,\dots,2^n\}$ of interior points that are needed and solve

$$\exists \boldsymbol{x}_1,\dots,\boldsymbol{x}_m \in \Pi \cap \mathbb{Z}^d \quad \text{and} \quad \lambda_{\boldsymbol{x}} \in \mathbb{Z}_{\geq 0}\ \forall \boldsymbol{x} \in \operatorname{vert}(\Pi) \quad \text{with} \quad \boldsymbol{a} = \sum_{\boldsymbol{x}\in\operatorname{vert}(\Pi)} \lambda_{\boldsymbol{x}} \cdot \boldsymbol{x} + \sum_{i=1}^{m} \boldsymbol{x}_i$$

This is an integer linear program with $d\cdot m + 2^d \leq (d+1)\cdot 2^d$ many variables. If we consider d as a constant, then this integer linear program can be solved in polynomial time.

6.3.2 Covering polytopes with parallelepipeds

So, we are able to solve INTEGER CONIC PROGRAMMING for parallelepipeds, but one might argue that these are very special polytopes and the result is of limited interest. Somewhat surprisingly, one can more or less reduce the general case to the one for parallelepipeds. This is possible as every polytope can be covered with a bounded number of parallelepipeds.

Lemma 6.5. *Let $P = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ be a polytope with $\boldsymbol{A} \in \mathbb{Z}^{m\times d}$, $\boldsymbol{b} \in \mathbb{Z}^m$ and $\Delta := \max\{\|\boldsymbol{A}\|_\infty, \|\boldsymbol{b}\|_\infty\}$. Then there exists a set $\boldsymbol{\Pi}$ of at most $|\boldsymbol{\Pi}| \leq N := m^d d^{O(d)} (\log\Delta)^d$ many integral parallelepipeds such that*

$$P \cap \mathbb{Z}^d \subseteq \bigcup_{\Pi\in\boldsymbol{\Pi}} \Pi \subseteq P.$$

[2] Well, an algorithm cannot really "guess" anything. But one can simply try out all combinations; we know that at least one will work.

Moreover the set $\mathbf{\Pi}$ *can be computed in time* $N^{O(1)}$.

Proof. First of all, one can use *Cramer's rule* to upper bound the coordinates for every point $\boldsymbol{x} \in P$ by $\|\boldsymbol{x}\|_\infty \leq d! \cdot \Delta^d$. For each point $\boldsymbol{x} \in P$, this upper bounds the slack with respect to contraint i by $|\boldsymbol{A}_i \boldsymbol{x} - b_i| \leq (d+1)\Delta \cdot d! \cdot \Delta^d \leq (d+1)! \cdot \Delta^{d+1}$. We begin by partitioning the polytope P into *cells* in a similar way as we did it in Theorem 3.3 just that the cells are going to be finer by a factor of roughly d. We want to partition the interval $[0, (d+1)! \cdot \Delta^{d+1}]$ into smaller intervals $[\alpha_j, \alpha_{j+1}]$ such that for any integer values $p, q \in [\alpha_j, \alpha_{j+1}] \cap \mathbb{Z}$ one has $\frac{p}{q} \leq 1 + \frac{1}{d}$. For this we can choose $\alpha_j := (1 + \frac{1}{d})^{j-2}$ for $j = 1, \ldots, K$ and $\alpha_0 := 0$. It is not difficult to see that $K \leq O(d^3(\log \Delta + \log d))$ such intervals suffice.

Our next step is to partition P into *cells* such that points in the same cell have roughly the same slacks for all the constraints. For each sequence $j_1, \ldots, j_m \in \{1, \ldots, K\}$ we define a cell $C = C(j_1, \ldots, j_m)$ as

$$\left\{ \boldsymbol{x} \in \mathbb{R}^d \mid \alpha_{j_i} \leq b_i - \boldsymbol{A}_i \boldsymbol{x} \leq \alpha_{j_i+1} \; \forall i \in [m] \right\}.$$

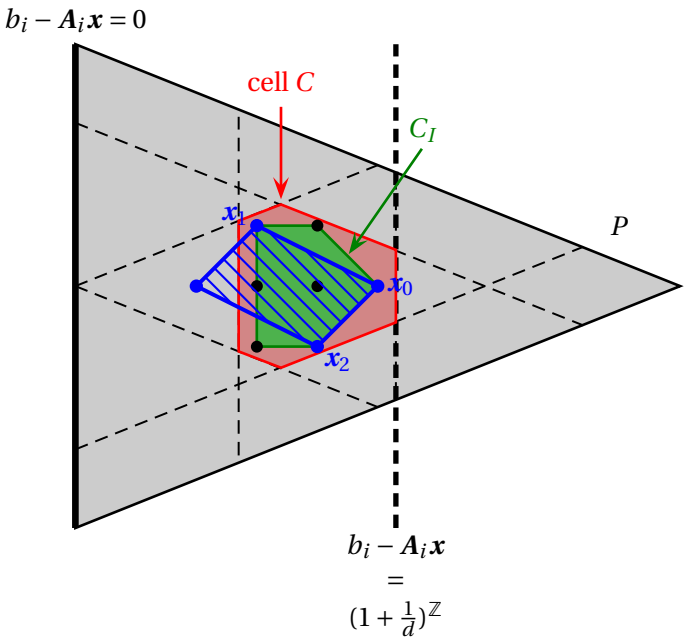
Fix one of those non-empty cells $C \subseteq P$ and let $\{\boldsymbol{x}_j\}_{j \in J}$ be the set of vertices of the integral hull $C_I := \operatorname{conv}\{C \cap \mathbb{Z}^d\}$. We will show that there are only $|\operatorname{vert}(C_I)|^{d+1}$ parallelepipeds necessary to cover the integer points of this cell (we assume that $C \cap \mathbb{Z}^d \neq \emptyset$, otherwise there is nothing to do). By Carathéodory's Theorem, any point $\boldsymbol{x} \in C_I$ lies already in the convex hull of at most $d+1$ affinely independent vertices of C_I. That means

$$C_I \subseteq \bigcup_{I \subseteq J : |I| \leq d+1} \operatorname{conv}(\{\boldsymbol{x}_i \mid i \in I\})$$

But the set $\operatorname{conv}(\{\boldsymbol{x}_i \mid i \in I\})$ is a simplex — not a parallelepiped. But we can simply extend the simplex to a parallelepiped. Suppose that $I = \{0, \ldots, k\}$, then we can define the integral parallelepiped

$$\Pi(I) := \left\{ \boldsymbol{x}_0 + \sum_{j \in J} \mu_j (\boldsymbol{x}_j - \boldsymbol{x}_0) \mid 0 \leq \mu_j \leq 1 \; \forall j \in J \right\}$$

Note that $\operatorname{conv}\{\boldsymbol{x}_i \mid i \in I\} \subseteq \Pi(I)$. In other words, $\Pi(I)$ for $I \subseteq J$ with $|I| \leq d+1$ does cover C_I. It remains to show that $\Pi(I) \subseteq P$ (note that $\Pi(I)$ is not necessarily contained in C_I).



Consider any point $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{j\in J} \mu_j(\boldsymbol{x}_j - \boldsymbol{x}_0)$ with $0 \le \mu_j \le 1$, where all $\boldsymbol{x}_j$'s are vertices of the same cell C_I. Then for any constraint $i \in [m]$, we have

$$b_i - \boldsymbol{A}_i\boldsymbol{x} \ge \underbrace{b_i - \boldsymbol{A}_i\boldsymbol{x}_0}_{\ge \alpha_{j_i}} - \sum_{j\in J} \underbrace{|\mu_j|}_{\le 1} \cdot \underbrace{|\boldsymbol{A}_i\boldsymbol{x}_j - \boldsymbol{A}_i\boldsymbol{x}_0|}_{\le \alpha_{j_i+1} - \alpha_{j_i} \le \frac{\alpha_{j_i}}{d}} \ge 0.$$

Note that the argument is constructive in the sense that the proof essentially gives an algorithm whose running time is only a polynomial factor larger than the bound that we have on the number of parallelepipeds. That concludes the proof. □

6.3.3 The algorithm

We now come to the main result where we show how to solve CONIC INTEGER PROGRAMMING for arbitrary polytopes P. The algorithm first computes a covering of P with integral parallelepipeds. Then it "guesses" that 2^d many that are needed in the unknown solution. Then we can just write down an integer linear program with $2^{O(d)}$ variables that finds the correct integer conic combination of the target vector $\boldsymbol{a}$.

Theorem 6.6. *Given a polytope P and a target vector $\boldsymbol{a}$, one find an integer conic combination $\boldsymbol{\lambda} \in \mathbb{Z}_{\ge 0}^{P\cap\mathbb{Z}^d}$ such that $\boldsymbol{a} = \sum_{\boldsymbol{x}\in P\cap\mathbb{Z}^d} \lambda_{\boldsymbol{x}}\boldsymbol{x}$ in time $enc(P)^{2^{O(d)}} \cdot enc(\boldsymbol{a})^{O(1)}$, or decide that no such combination exists. Moreover, the support of $\boldsymbol{\lambda}$ is always bounded by 2^{2d+1}.*

Proof of Main Theorem 6.6. Let $P = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}\}$ be the given polytope. Here we assume that the coefficients in the inequality description are integral and the numbers in $\boldsymbol{A}, \boldsymbol{b}$ are bounded in absolute value by Δ.

We use Lemma 6.5 to compute parallelepipeds covering P. Let X be the set of all vertices of those parallelepipeds. Note that $|X| \le N$ with $N := m^d d^{O(d)} (\log\Delta)^d$ and the time for this computation is of the form $N^{O(1)}$. We know by Lemma 6.4 that there is a vector $\boldsymbol{\lambda}^* \in \mathbb{Z}_{\ge 0}^{P\cap\mathbb{Z}^d}$ such that $\sum_{\boldsymbol{x}\in P\cap\mathbb{Z}^d} \lambda_{\boldsymbol{x}}\boldsymbol{x} = \boldsymbol{a}$, $|\mathrm{supp}(\boldsymbol{\lambda}^*) \cap X| \le 2^d$, $|\mathrm{supp}(\boldsymbol{\lambda}^*) \setminus X| \le 2^d$ and $\lambda^*_{\boldsymbol{x}} \in \{0,1\}$ for $\boldsymbol{x} \in (P\cap\mathbb{Z}^d)\setminus X$.

At the expense of a factor $N^{2^{2d}}$ we guess the subset $X' = X \cap \mathrm{supp}(\boldsymbol{\lambda}^*)$[3]. At the expense of another factor $2^{2d}+1$ we guess the number $k = \sum_{\boldsymbol{x}\notin X'} \lambda^*_{\boldsymbol{x}} \in \{0,\dots,2^{2d}\}$ of extra points. Now we can set up an integer program with few variables. We use variables $\lambda_{\boldsymbol{x}}$ for $\boldsymbol{x} \in X'$ to determine the correct multiplicities of the points in X. Moreover, we have variables $\boldsymbol{x}_1,\dots,\boldsymbol{x}_k \in \mathbb{Z}^d_{\ge 0}$ to determine which extra points to take with unit weight. The ILP is then of the form

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x}_i &\le \boldsymbol{b} \quad \forall i = 1,\dots,k \\ \sum_{\boldsymbol{x}\in X'} \lambda_{\boldsymbol{x}}\boldsymbol{x} + \sum_{i=1}^{k} \boldsymbol{x}_i &= \boldsymbol{a} \\ \lambda_{\boldsymbol{x}} &\in \mathbb{Z}_{\ge 0} \quad \forall \boldsymbol{x} \in X' \\ \boldsymbol{x}_i &\in \mathbb{Z}^d \quad \forall i = 1,\dots,k \end{aligned}$$

and given that we made the guessing correctly, this system has a solution. The number of variables is $|X'| + (k+1)d \le 2^{O(d)}$ and the number of constraints is $km + d + |X'|d = 2^{O(d)}m$ as well. Note that the

[3] Actually we know that X' consists of the vertices of at most 2^d parallelepipeds, thus it suffices to incorporate a factor of N^{2^d}, but the improvement would be absorbed by the O-notation later, anyway.

largest coefficient is at most $\Delta' = d! \cdot \Delta^d$. Hence the system can be solved[4] in time $(2^{O(d)})^{2^{O(d)}} \cdot (2^{O(d)} m)^{O(1)} \cdot (\log \Delta')^{O(1)}$ via Theorem 2.7. The total running time is hence of the form $\text{enc}(P)^{2^{O(d)}}$. □

6.4 Exercises

Exercise 6.1. For any k, give a set $X \subseteq \mathbb{R}$ of size $|X| = k$ and a vector $b \in \text{int.cone}(X)$ so that for any strict subset $Y \subset X$ one has $b \notin \text{int.cone}(Y)$.

Exercise 6.2. Fix a dimension $d \geq 2$ and let $k := 2^{d-1}$. Let's define a set $X \subset \mathbb{Z}^d$ of k points as

$$\left\{(1 + x_1, \ldots, 1 + x_{d-1}, (4k)^{1+\sum_{i=1}^{d-1} 2^{i-1} x_i}) \mid x_i \in \{0,1\}\right\}.$$

We sort $X = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\}$ according to their length, i.e. $\|\boldsymbol{a}_i\|_\infty = (4k)^i$ and define $P := \text{conv}(X)$. Show that there is one vector $\boldsymbol{b} \in \text{int.cone}(X)$ so that for all subset $Y \subseteq P \cap \mathbb{Z}^d$ with $|Y| < 2^{d-1}$ one has $\boldsymbol{b} \notin \text{int.cone}(Y)$.
Hint: Use $\boldsymbol{b} := \boldsymbol{a}_1 + \ldots + \boldsymbol{a}_k$.

Exercise 6.3. Let $a_1, \ldots, a_n \in [0,1]$ be numbers. Use the *pigeonhole principle* to show that there is a $\boldsymbol{x} \in \{-1,0,1\}^n \setminus \{\boldsymbol{0}\}$ so that $|\sum_{i=1}^n x_i a_i| \leq \frac{n}{2^n - 1}$.
Remark: The proof will be non-constructive. The best known bound that can be found with a polynomial time algorithm satisfies only $|\sum_{i=1}^n x_i a_i| \leq n^{-\Theta(\log n)}$. This can be done using Karmarkar and Karp's "differencing algorithm".

[4] Actually instead of using the time $2^{O(n^3)}$ of Lenstra's original algorithm, this calculation uses the $n^{O(n)}$ improvement of Kannan to solve integer programming in n variables.

Chapter 7

Banaszczyk's Transference Theorem

We want to go back to lattices. Recall that for a lattice Λ we denoted $\Lambda^* := \{\boldsymbol{y} \in \mathbb{R}^n \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}\ \forall \boldsymbol{x} \in \Lambda\}$ as the *dual lattice*. Also recall that $\lambda_i(\Lambda)$ gives the minimum value so that there are i linearly independent vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i \in \Lambda$ with $\|\boldsymbol{v}_1\|_2, \ldots, \|\boldsymbol{v}_i\|_2 \leq \lambda_i(\Lambda)$. We have seen in Exercise 2.5 that lattice basis reduction implies that one always has $\lambda_1(\Lambda) \cdot \lambda_n(\Lambda^*) \leq 2^{O(n^2)}$, which already is a remarkable statement in the sense that knowing the length $\lambda_1(\Lambda)$ of a single vector in the primal lattice gives an upper bound on n vectors in the dual lattice. However, the bound is exponentially large and hence quite weak. It turns out that using Fourier analysis one can prove dramatically better bounds:

Theorem 7.1 (Banaszczyk '93 [Ban93b])**.** *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$ one has $1 \leq \lambda_1(\Lambda) \cdot \Lambda_n(\Lambda^*) \leq n$.*

Recall that the lower bound follows from the definition of the dual lattice. We also saw that Minkowski's theorem implies that $\lambda_1(\Lambda) \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}$ and $\lambda_1(\Lambda^*) \leq \sqrt{n} \cdot \det(\Lambda^*)^{1/n}$ which then gives the qualitatively weaker bound of $\lambda_1(\Lambda) \cdot \lambda_1(\Lambda^*) \leq n$. The technique used for proving Banaszczyk's Theorem are fundamentally different from the techniques we have seen so far. They rely on *Fourier analysis* and the *Discrete Gaussian*. This chapter is a reproduction of the fantastic lecture notes of Regev [Reg09].

7.1 Fourier analysis

The idea behind Fourier analysis is to express a function f in a different basis (the Fourier basis). Many insights can be derived from this view that are hidden otherwise.

7.1.1 The Fourier Transform

A classical object of study in functional analysis is the Fourier transform.

Definition 6. For a function $f : \mathbb{R}^n \to \mathbb{C}$ with $\int_{\mathbb{R}^n} |f(\boldsymbol{x})|\, d\boldsymbol{x} < \infty$ we define the *Fourier transform* as the function $\hat{f} : \mathbb{R}^n \to \mathbb{C}$ with

$$\hat{f}(\boldsymbol{y}) := \int_{\mathbb{R}^n} f(\boldsymbol{x}) \cdot e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle}\, d\boldsymbol{x}$$

It is not hard to see that some technical conditions on function f are needed. In the proof of Banaszczyk's Theorem we will apply the Fourier transform only to a family of functions f that are continuous and decay exponentially. Hence we will never run into any convergence problem. Let us call

such functions "*nice enough*" without making this more formal. We will not go into any detail which conditions on functions are necessary and which are sufficient.

A popular view is to consider the function $f(\boldsymbol{x})$ as a "signal" and the Fourier coefficient $\hat{f}(\boldsymbol{y})$ gives the amplitudes of the "frequency" $\boldsymbol{y}$ of that signal. There is also an explicit way to assemble the "frequencies" to recover the "signal".

Theorem 7.2 (Inversion Formula for the Fourier Transform)**.** *For a nice enough function $f : \mathbb{R}^n \to \mathbb{C}$ one has*

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^n} \hat{f}(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} \, d\boldsymbol{y} \quad \forall \boldsymbol{x} \in \mathbb{R}^n.$$

We will not actually need this theorem and hence skip its proof.

7.1.2 The Fourier series

We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is Λ-*periodic* if $f(\boldsymbol{x}) = f(\boldsymbol{x} + \boldsymbol{y})$ for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \Lambda$. In other words, shifting the argument of f by a lattice point leaves the value invariant. For example, the function $\operatorname{dist}(\boldsymbol{x}, \Lambda) := \min\{\|\boldsymbol{x} - \boldsymbol{y}\|_2 : \boldsymbol{y} \in \Lambda\}$ that gives the distance of a point to the nearest lattice point is a natural Λ-periodic function. Now we want to define a discrete version of the Fourier transform:

Definition 7. Let $\Lambda = \Lambda(\boldsymbol{B}) \subseteq \mathbb{R}^n$ be a full-rank lattice and let $f : \mathbb{R}^n \to \mathbb{C}$ be a Λ-periodic function. Define the *Fourier series* $\tilde{f} : \Lambda^* \to \mathbb{C}$ as

$$\tilde{f}(\boldsymbol{y}) := \frac{1}{\det(\Lambda)} \cdot \int_{\mathcal{P}(\boldsymbol{B})} f(\boldsymbol{x}) \cdot e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} \, d\boldsymbol{x} \quad \forall \boldsymbol{y} \in \Lambda^*.$$

Observe that the definition itself includes a concrete basis $\boldsymbol{B}$ for the lattice. We leave it as an exercise to prove that the values $\tilde{f}(\boldsymbol{y})$ do not depend on the chosen basis.

Now we come to the discrete analogue of Theorem 7.2. We want to at least sketch the proof — but we will be quite informal.

Theorem 7.3 (Inversion Formula for Lattices)**.** *Let $\Lambda = \Lambda(\boldsymbol{B}) \subseteq \mathbb{R}^n$ be a full-rank lattice and let $f : \mathbb{R}^n \to \mathbb{C}$ be a nice enough Λ-periodic function and let $\tilde{f} : \Lambda^* \to \mathbb{C}$ be its Fourier series. Then*

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \Lambda^*} \tilde{f}(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} \quad \forall \boldsymbol{x} \in \mathbb{R}^n$$

Proof sketch. Let us sketch the statement for the integer lattice $\Lambda = \mathbb{Z}^n$. Fix a vector $\boldsymbol{x} \in \mathbb{R}^n$. Then we can write

$$\begin{aligned}
\sum_{\boldsymbol{y} \in \mathbb{Z}^n} \tilde{f}(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} &\overset{\text{Def. } \tilde{f}}{=} \sum_{\boldsymbol{y} \in \mathbb{Z}^n} \left(\int_{[0,1[^n} f(\boldsymbol{z}) \cdot e^{-2\pi i \langle \boldsymbol{z}, \boldsymbol{y} \rangle} \, d\boldsymbol{z} \right) \cdot e^{2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} \\
&= \sum_{\boldsymbol{y} \in \mathbb{Z}^n} \left(\int_{[0,1[^n} f(\boldsymbol{z}) \cdot e^{2\pi i \langle \boldsymbol{x} - \boldsymbol{z}, \boldsymbol{y} \rangle} \, d\boldsymbol{z} \right) \\
&= \int_{[0,1[^n} f(\boldsymbol{z}) \cdot \underbrace{\left(\sum_{\boldsymbol{y} \in \mathbb{Z}^n} e^{2\pi i \langle \boldsymbol{x} - \boldsymbol{z}, \boldsymbol{y} \rangle} \right)}_{= 0 \text{ if } \boldsymbol{x} - \boldsymbol{z} \notin \mathbb{Z}^n;\ = \infty \text{ if } \boldsymbol{x} - \boldsymbol{z} \in \mathbb{Z}^n} d\boldsymbol{z} = f(\boldsymbol{x}).
\end{aligned}$$

The last step can and should be made more formal. But the intuition is that there is only a "peak" in the integral if $\boldsymbol{x} - \boldsymbol{z} \in \mathbb{Z}^n$. To see this better one could study the 1-dimensional case and consider the function

$F(k) := \sum_{y\in\{-k,\dots,k\}} e^{2\pi i\alpha\cdot y}$. Then for $\alpha\in\mathbb{Z}$ one has $F(k) = 2k+1$ and hence $\lim_{k\to\infty} F(k) = \infty$. But for any $\alpha\notin\mathbb{Z}$ one has $|F(k)| = o(k)$. □

The Poisson Summation Formula shows that the sum of a function f over all lattice points is the same as the sum over the Fourier transform $\hat{f}$ over all points in the dual lattice (up to normalization factor).

Theorem 7.4 (Poisson Summation Formula for Lattices)**.** *For a nice enough function $f:\mathbb{R}^n\to\mathbb{R}$ and a full-rank lattice $\Lambda\subseteq\mathbb{R}^n$ one has $f(\Lambda) = \det(\Lambda^*)\cdot\hat{f}(\Lambda^*)$.*

Proof. We want to apply the Inversion Formula for Lattices, which holds for Λ-periodic functions. But the function $\varphi(\boldsymbol{x}) := \sum_{\boldsymbol{z}\in\Lambda} f(\boldsymbol{x}+\boldsymbol{z})$ is Λ-periodic by definition. Then for any $\boldsymbol{y}\in\Lambda^*$ we have

$$\begin{aligned}
\tilde{\varphi}(\boldsymbol{y}) &\overset{\text{Def. Fourier series}}{=} \frac{1}{\det(\Lambda)}\int_{\mathcal{P}(\boldsymbol{B})}\varphi(\boldsymbol{x})\cdot e^{-2\pi i\langle\boldsymbol{x},\boldsymbol{y}\rangle}\,d\boldsymbol{x} \qquad (*)\\
&\overset{\text{Def. }\varphi}{=} \frac{1}{\det(\Lambda)}\int_{\mathcal{P}(\boldsymbol{B})}\sum_{\boldsymbol{z}\in\Lambda} f(\boldsymbol{x}+\boldsymbol{z})\cdot e^{-2\pi i\langle\boldsymbol{x},\boldsymbol{y}\rangle}\,d\boldsymbol{x}\\
&\overset{\text{swapping order}}{=} \frac{1}{\det(\Lambda)}\sum_{\boldsymbol{z}\in\Lambda}\int_{\mathcal{P}(\boldsymbol{B})} f(\boldsymbol{x}+\boldsymbol{z})\cdot e^{-2\pi i\langle\boldsymbol{x},\boldsymbol{y}\rangle}\,d\boldsymbol{x}\\
&\overset{\boldsymbol{z}\in\Lambda\Rightarrow e^{-2\pi i\langle\boldsymbol{z},\boldsymbol{y}\rangle}=1}{=} \frac{1}{\det(\Lambda)}\sum_{\boldsymbol{z}\in\Lambda}\int_{\mathcal{P}(\boldsymbol{B})} f(\boldsymbol{x}+\boldsymbol{z})\cdot e^{-2\pi i\langle\boldsymbol{x}+\boldsymbol{z},\boldsymbol{y}\rangle}\,d\boldsymbol{x}\\
&\overset{\Lambda+\mathcal{P}(\boldsymbol{B})=\mathbb{R}^n}{=} \underbrace{\frac{1}{\det(\Lambda)}}_{=\det(\Lambda^*)}\underbrace{\int_{\mathbb{R}^n} f(\boldsymbol{x})\cdot e^{-2\pi i\langle\boldsymbol{x},\boldsymbol{y}\rangle}\,d\boldsymbol{x}}_{=\hat{f}(\boldsymbol{y})} = \det(\Lambda^*)\cdot\hat{f}(\boldsymbol{y}).
\end{aligned}$$

The Inversion Formula for Lattices from Theorem 7.3 now says that

$$f(\Lambda) = \sum_{\boldsymbol{z}\in\Lambda} f(\boldsymbol{0}+\boldsymbol{z}) \overset{\text{Def. }\varphi}{=} \varphi(\boldsymbol{0}) \overset{\text{Inversion Formula for Lattices}}{=} \sum_{\boldsymbol{y}\in\Lambda^*}\tilde{\varphi}(\boldsymbol{y})\cdot\underbrace{e^{2\pi i\langle\boldsymbol{0},\boldsymbol{y}\rangle}}_{=1} \overset{(*)}{=} \sum_{\boldsymbol{y}\in\Lambda^*}\underbrace{\det(\Lambda^*)\hat{f}(\boldsymbol{y})}_{=\tilde{\varphi}(\boldsymbol{y})} = \det(\Lambda^*)\cdot\hat{f}(\Lambda^*).$$

□

7.2 The discrete Gaussian

A crucial function in the proof of Banaszczyk's Theorem is the *discrete Gaussian* which is a function

$$\rho_s:\mathbb{R}^n\to\mathbb{R}_{\geq 0} \quad\text{with}\quad \rho_s(\boldsymbol{x}) := e^{-\pi\|\boldsymbol{x}/s\|_2^2} \quad \forall\boldsymbol{x}\in\mathbb{R}^n.$$

In particular we will consider the sum $\rho_s(\Lambda)$ over a lattice. Intuitively, the quantity $\rho_s(\Lambda)$ counts the lattice points while the contribution of each point $\boldsymbol{x}\in\Lambda$ fades quickly if $\|\boldsymbol{x}\|_2$ is getting too large.

First we prove that for $s=1$, the Fourier transform of the discrete Gaussian is again the discrete Gaussian:

Lemma 7.5. *For all $s>0$, the Fourier transform of the discrete Gaussian is $\hat{\rho}_s(\boldsymbol{x}) = s^n\cdot\rho_{1/s}(\boldsymbol{x})$ for all $\boldsymbol{x}\in\mathbb{R}^n$.*

Proof. Let us define the coordinate contribution as $g_s(x) := e^{-\pi\cdot(x/s)^2}$ for $x\in\mathbb{R}$. The following two facts can be proven using the proper integral manipulation skills. We will skip the proof here:

Fact I. One has $\hat{g}_s(y) = s \cdot e^{-\pi \cdot (ys)^2}$ for all $y \in \mathbb{R}$.

Fact II. For any function $f : \mathbb{R}^n \to \mathbb{R}$ in product form $f(\boldsymbol{x}) = \prod_{i=1}^n f_i(x_i)$ one has $\hat{f}(\boldsymbol{y}) = \prod_{i=1}^n \hat{f}_i(y_i)$ for $\boldsymbol{y} \in \mathbb{R}^n$.

Next, observe that the discrete Gaussian has a product structure and we can write

$$\rho_s(\boldsymbol{x}) = e^{-\pi \|\boldsymbol{x}/s\|_2^2} = \prod_{i=1}^n e^{-\pi (x_i/s)^2} = \prod_{i=1}^n g_s(x_i).$$

Then for $\boldsymbol{y} \in \mathbb{R}^n$ the Fourier transform is

$$\hat{\rho}_s(\boldsymbol{y}) \overset{\text{Fact II}}{=} \prod_{i=1}^n \hat{g}(y_i) \overset{\text{Fact I}}{=} \prod_{i=1}^n (s \cdot e^{-\pi \cdot (sy_i)^2}) = s^n \cdot e^{-\pi \| \frac{1}{1/s} \boldsymbol{y} \|_2^2} = s^n \cdot \rho_{1/s}(\boldsymbol{y}).$$

□

This implies a useful relation between the sum of the discrete Gaussian over a lattice and its dual lattice:

Corollary 1. *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$ and any $s > 0$ one has*

$$\rho_s(\Lambda) = \det(\Lambda^*) \cdot s^n \cdot \rho_{1/s}(\Lambda^*).$$

Proof. Follows from Lemma 7.5 and the Poisson Summation Formula for Lattices from Lemma 7.4. □

It is not hard to exactly quantify the sum of the discrete Gaussian over a *shifted* lattice as well. Essentially if we shift the lattice by $\boldsymbol{u}$, then we need to "pull out" a factor of $e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle}$ for every summand.

Corollary 2. *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$, any $s > 0$ and $\boldsymbol{u} \in \mathbb{R}^n$ one has*

$$\rho_s(\Lambda + \boldsymbol{u}) = \det(\Lambda^*) \cdot s^n \cdot \sum_{\boldsymbol{y} \in \Lambda^*} \rho_{1/s}(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle}.$$

Proof. We consider the function $f(\boldsymbol{x}) := \rho_s(\boldsymbol{x} + \boldsymbol{u})$ and write the Fourier transform as

$$\hat{f}(\boldsymbol{x}) = \int_{\mathbb{R}^n} \rho_s(\boldsymbol{y} + \boldsymbol{u}) e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} \, d\boldsymbol{y} = \int_{\mathbb{R}^n} \rho_s(\boldsymbol{y}) e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{y} - \boldsymbol{u} \rangle} \, d\boldsymbol{y} = \hat{\rho}_s(\boldsymbol{x}) \cdot e^{2\pi i \langle \boldsymbol{x}, \boldsymbol{u} \rangle} \qquad (*)$$

for $\boldsymbol{x} \in \mathbb{R}^n$. Then applying Lemma 7.4 to f gives

$$\begin{aligned} \rho_s(\Lambda + \boldsymbol{u}) &= f(\Lambda) \overset{\text{Lemma 7.4}}{=} \det(\Lambda^*) \cdot \sum_{\boldsymbol{y} \in \Lambda^*} \hat{f}(\boldsymbol{y}) \overset{(*)}{=} \det(\Lambda^*) \cdot \sum_{\boldsymbol{y} \in \Lambda^*} \hat{\rho}_s(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle} \\ &\overset{\hat{\rho}_s(\boldsymbol{y}) = s^n \rho_{1/s}(\boldsymbol{y})}{=} \det(\Lambda^*) \cdot s^n \cdot \sum_{\boldsymbol{y} \in \Lambda^*} \rho_{1/s}(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle}. \end{aligned}$$

□

An important insight is that shifting the lattice can only *decrease* the sum over the discrete Gaussian:

Lemma 7.6. *Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice, $s > 0$ and let $\boldsymbol{u} \in \mathbb{R}^n$ be a shift. Then*

$$\rho_s(\Lambda + \boldsymbol{u}) \leq \rho_s(\Lambda).$$

Proof. We can estimate that

$$\begin{aligned}
\rho_s(\Lambda + \boldsymbol{u}) &\overset{\text{Cor. 2}}{=} \left| \det(\Lambda^*) \cdot s^n \cdot \sum_{\boldsymbol{y} \in \Lambda^*} \rho_{1/s}(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle} \right| \\
&\leq \det(\Lambda^*) \cdot s^n \cdot \sum_{\boldsymbol{y} \in \Lambda^*} \rho_{1/s}(\boldsymbol{y}) \cdot \underbrace{|e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle}|}_{\leq 1} \\
&\leq \det(\Lambda^*) \cdot s^n \cdot \underbrace{\sum_{\boldsymbol{y} \in \Lambda^*} \rho_{1/s}(\boldsymbol{y})}_{= \rho_{1/s}(\Lambda^*)} \\
&\overset{\text{Cor. 1}}{=} \rho_s(\Lambda)
\end{aligned}$$

which gives the claim. □

Increasing the scaling factor s for the discrete Gaussians means that the effective length of the lattice vectors is reduced and the sum $\rho_s(\Lambda)$ would increase. But we can limit the decrease and show that it cannot be more than exponential.

Lemma 7.7. *Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice and let $\boldsymbol{u} \in \mathbb{R}^n$ and $s \geq 1$. Then $\rho_s(\Lambda + \boldsymbol{u}) \leq s^n \cdot \rho_1(\Lambda)$.*

Proof. It suffices to prove that $\rho_s(\Lambda) \leq s^n \cdot \rho_1(\Lambda)$ — the general claim follows then from Lemma 7.6 where we showed that shifting can only decrease the sum of the discrete Gaussian. We will use the formula from Cor. 1 twice and obtain

$$\rho_s(\Lambda) \overset{\text{Cor.1 for s}}{=} \det(\Lambda^*) \cdot s^n \cdot \sum_{\boldsymbol{y} \in \Lambda^*} \underbrace{\rho_{1/s}(\boldsymbol{y})}_{\leq \rho_1(\boldsymbol{y})} \leq \det(\Lambda^*) \cdot s^n \sum_{\boldsymbol{y} \in \Lambda^*} \rho_1(\boldsymbol{y}) \overset{\text{Cor. 1 for } s=1}{=} s^n \cdot \rho_1(\Lambda).$$

This gives the claim. □

7.3 The Proof of Banaszczyk's Theorem

Finally we come to the main part of proving the Transference Theorem of Banaszczyk's. Instead of using the successive minimum $\lambda_n(\Lambda^*)$ of the dual lattice directly, we will use the covering radius:

Definition 8. For a full-rank lattice $\lambda \subseteq \mathbb{R}^n$, the *covering radius* is the maximum distance of any point to the lattice, that is

$$\mu(\Lambda) := \max_{\boldsymbol{x} \in \mathbb{R}^n} \operatorname{dist}(\boldsymbol{x}, \Lambda)$$

where $\operatorname{dist}(\boldsymbol{x}, \Lambda) := \min\{\|\boldsymbol{x} - \boldsymbol{y}\|_2 \mid \boldsymbol{y} \in \Lambda\}$.

We have seen the following fact already in Exercise 1.5:

Corollary 3. *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$ one has $\mu(\Lambda) \geq \frac{1}{2}\Lambda_n(\Lambda)$.*

Note that it is possible that $\mu(\Lambda) \gg \lambda_n(\Lambda)$. For example for the integer lattice $\mathbb{Z}^n$ one has $\lambda_n(\mathbb{Z}^n) = 1$ while $\mu(\mathbb{Z}^n) = \frac{1}{2}\sqrt{n}$ (for any smaller radius the point $(\frac{1}{2}, \ldots, \frac{1}{2})$ would not be covered). However, we will show that $1 \leq \lambda_1(\Lambda) \cdot \mu(\Lambda^*) \leq n$ which then gives the slightly weaker bound of $1 \leq \lambda_1(\Lambda) \cdot \Lambda_n(\Lambda^*) \leq 2n$.

Recall that $\mathcal{B}(\boldsymbol{c}, r) := \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x} - \boldsymbol{c}\|_2 \leq r\}$ is the ball of radius r around the center $\boldsymbol{c}$. If we consider the sum $\rho_1(\Lambda) = \sum_{\boldsymbol{x} \in \Lambda} e^{-\pi \|\boldsymbol{x}\|_2^2}$ then we know that always $\rho_1(\Lambda) \geq e^{-\pi \|\boldsymbol{0}\|_2^2} = 1$ due to the contribution of the origin. A useful insight is that the contribution of lattice points outside of a ball of radius $\sqrt{n}$ to $\rho_1(\Lambda)$ is always negligible:

Lemma 7.8. *For any full-rank lattice $\Lambda \subseteq \mathbb{R}^n$ and any vector $\boldsymbol{u} \in \mathbb{R}^n$ one has*

$$\rho_1((\Lambda + \boldsymbol{u}) \setminus \mathcal{B}(\mathbf{0}, \sqrt{n})) \leq 2^{-n} \cdot \rho_1(\Lambda).$$

Proof. The proof basically uses that for long vectors the value $\rho_s(\boldsymbol{x})$ increases a lot with s, while we know that the overall sum can only grow with s^n. Then clearly long vectors could not have contributed much to the sum. More formally

$$\begin{aligned}
2^n \cdot \rho_1(\Lambda) &\overset{\text{Lem. 7.7}}{\geq} \rho_2(\Lambda + \boldsymbol{u}) \\
&\geq \rho_2((\Lambda + \boldsymbol{u}) \setminus \mathcal{B}(\mathbf{0}, \sqrt{n})) \\
&= \sum_{\boldsymbol{y} \in \Lambda + \boldsymbol{u} : \|\boldsymbol{y}\|_2 > \sqrt{n}} e^{-\pi \|\boldsymbol{y}/2\|_2^2} \\
&= \sum_{\boldsymbol{y} \in \Lambda + \boldsymbol{u} : \|\boldsymbol{y}\|_2 > \sqrt{n}} \underbrace{e^{\frac{3}{4}\pi \|\boldsymbol{y}\|_2^2}}_{\geq 4^n} \cdot e^{-\pi \|\boldsymbol{y}\|_2^2} \\
&\geq 4^n \cdot \rho_1((\Lambda + \boldsymbol{u}) \setminus \mathcal{B}(\mathbf{0}, \sqrt{n}))
\end{aligned}$$

Rearranging then gives the claim. □

An easy consequence is that in a lattice without short vectors, essentially all the Gaussian weight has to lie on the origin $\mathbf{0}$:

Lemma 7.9. *Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice with $\lambda_1(\Lambda) > \sqrt{n}$. Then $\rho_1(\Lambda \setminus \{\mathbf{0}\}) \leq 2 \cdot 2^{-n}$.*

Proof. Using the previous Lemma we have

$$\rho_1(\Lambda \setminus \{\mathbf{0}\}) \overset{\Lambda_1(\Lambda) > \sqrt{n}}{=} \rho_1(\Lambda \setminus \mathcal{B}(\mathbf{0}, \sqrt{n})) \overset{\text{Lem. 7.8}}{\leq} 2^{-n} \cdot \rho_1(\Lambda) = 2^{-n} \cdot (\underbrace{\rho_1(\mathbf{0})}_{=1} + \rho_1(\Lambda \setminus \{\mathbf{0}\}))$$

Rearranging for $\rho_1(\Lambda \setminus \{\mathbf{0}\})$ then gives the claim. □

The next lemma gives one crucial insight: if the lattice Λ has no vector of length $\sqrt{n}$ or less, then the sum $\rho_1(\Lambda^* + \boldsymbol{u})$ over the shifted dual lattice does only marginally depend on the shift $\boldsymbol{u}$. This will then quickly imply that the dual lattice has no large "holes" and the covering radius has to be small.

Lemma 7.10. *Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice with $\Lambda_1(\Lambda) > \sqrt{n}$. Then for any vector $\boldsymbol{u} \in \mathbb{R}^n$ one has*

$$\rho_1(\Lambda^* + \boldsymbol{u}) = (1 \pm 2 \cdot 2^{-n}) \cdot \det(\Lambda).$$

Proof. We estimate that

$$|\rho_1(\Lambda^* + \boldsymbol{u}) - \det(\Lambda)| \overset{\text{Cor. 2}}{=} \det(\Lambda) \cdot \left| \sum_{\boldsymbol{y} \in \Lambda} \rho_1(\boldsymbol{y}) \cdot e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle} - 1 \right| \leq \det(\Lambda) \cdot \underbrace{\sum_{\boldsymbol{y} \in \Lambda \setminus \{\mathbf{0}\}} \rho_1(\boldsymbol{y})}_{\leq 2 \cdot 2^{-n} \text{ by Lem. 7.9}} \cdot \underbrace{|e^{2\pi i \langle \boldsymbol{y}, \boldsymbol{u} \rangle}|}_{\leq 1} \leq \det(\Lambda) \cdot 2 \cdot 2^{-n}$$

which gives the claim. □

Now we can prove Banaszczyk's result: we will show that for any lattice Λ with no non-zero vector of length at most $\sqrt{n}$, the covering radius of the dual lattice is bounded by $\sqrt{n}$.

Theorem 7.11. *For any full rank lattice $\Lambda \subseteq \mathbb{R}^n$ one has $\lambda_1(\Lambda) \cdot \mu(\Lambda^*) \leq n$.*

Proof. After scaling the lattice appropriately it suffices to assume $\lambda_1(\Lambda) > \sqrt{n}$ and $\mu(\Lambda^*) > \sqrt{n}$ and bring this to a contradiction. From Lemma 7.10 we know that for a lattice Λ with $\lambda_1(\Lambda) > \sqrt{n}$, shifting the dual lattice has little effect on the sum of the discrete Gaussian, that means $\rho_1(\Lambda^* - \boldsymbol{u}) \geq (1 - 2 \cdot 2^{-n}) \cdot \rho_1(\Lambda^*)$. Let $\boldsymbol{u} \in \mathbb{R}^n$ be the vector attaining the covering radius for the dual lattice, that means $\Lambda^* \cap \mathcal{B}(\boldsymbol{u}, \sqrt{n}) = \emptyset$. Then

$$\frac{1}{2}\rho_1(\Lambda^*) \overset{\text{Lem. 7.10}}{\leq} \rho_1(\Lambda^* - \boldsymbol{u}) \overset{\Lambda^* \cap \mathcal{B}(\boldsymbol{u}, \sqrt{n}) = \emptyset}{=} \rho_1((\Lambda^* - \boldsymbol{u}) \setminus \mathcal{B}(\boldsymbol{0}, \sqrt{n})) < 2^{-n}\rho_1(\Lambda^*)$$

which is a contradiction for $n \geq 2$. □

7.4 Exercises

Exercise 7.1. Prove that in any full-rank lattice Λ one has $\mu(\Lambda) \leq n \cdot \lambda_n(\Lambda)$.
Extra point: Prove that even $\mu(\Lambda) \leq O(\sqrt{n}) \cdot \lambda_n(\Lambda)$.

Exercise 7.2. Make the proof of Theorem 7.3 formal.

Exercise 7.3. Show that the definition of the Fourier series does not depend on the chosen basis. More precisely, let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice and let $f : \mathbb{R}^n \to \mathbb{R}$ be a Λ-periodic function. Suppose $\boldsymbol{B}_1, \boldsymbol{B}_2$ are basis with $\Lambda = \Lambda(\boldsymbol{B}_1) = \Lambda(\boldsymbol{B}_2)$. Prove that for for all $\boldsymbol{y} \in \Lambda^*$ one has

$$\int_{\mathcal{P}(\boldsymbol{B}_1)} f(\boldsymbol{x}) \cdot e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} \, d\boldsymbol{x} = \int_{\mathcal{P}(\boldsymbol{B}_2)} f(\boldsymbol{x}) \cdot e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{y} \rangle} \, d\boldsymbol{x}.$$

Chapter 8

Discrepancy Theory

Suppose we have a *set system* $\mathcal{S} = \{S_1, \ldots, S_m\}$ with sets $S_1, \ldots, S_m \subseteq [n]$, where we abbreviate $[n] := \{1, \ldots, n\}$ as the *elements*. A *coloring* is of the form $\chi : [n] \to \{-1, 1\}$, that means each element is either "colored" with -1 or with $+1$. The *discrepancy* of a coloring χ with respect to a set system $\mathcal{S}$ is the *maximum in-balance* $\text{disc}(\chi, \mathcal{S}) := \max_{S \in \mathcal{S}} |\sum_{j \in S} \chi(j)|$ of any set in the set system. The goal is to determine the discrepancy of the best coloring, that means

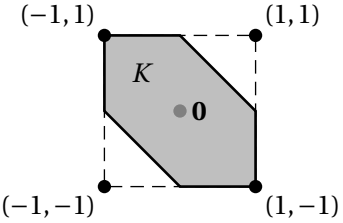
$$\text{disc}(\mathcal{S}) := \min_{\chi:[n]\to\{-1,1\}} \max_{S \in \mathcal{S}} |\chi(S)|$$

where we abbreviate $\chi(S) := \sum_{j \in S} \chi(j)$.

One might wonder, where is the connection to integer optimization and lattices. But for a set system $\mathcal{S}$ and a target parameter $\lambda \geq 0$ one could define a set

$$K := \left\{ \boldsymbol{x} \in \mathbb{R}^n : |\sum_{i \in S} x_i| \leq \lambda \ \forall S \in \mathcal{S} \right\}.$$

Then $\mathcal{S}$ has a coloring of discrepancy at most λ if and only if $K \cap \{-1, 1\}^n \neq \emptyset$. For example, for the $S = \{\{1\}, \{2\}, \{1, 2\}\}$ and $\lambda = 1$, the set K looks as follows:



Observe that K is a symmetric, convex set. Actually it is the intersection of $|\mathcal{S}|$ many *strips* where the strip $\{\boldsymbol{x} \in \mathbb{R}^n \mid |\sum_{i \in S} x_i| \leq \lambda\}$ induced by set $S \in \mathcal{S}$ has a geometric width of $\frac{2\lambda}{\sqrt{|S|}}$.

So indeed, discrepancy theory is the question of when a symmetric convex set contains a hypercube vertex. We will see that some arguments that we learned about lattices still apply here and other arguments apply that do not seem to work for lattices. A well written introduction into discrepancy with a lot of historical context can be found in Chapter 4 of Matousek [Mat99]. Our section 8.1 is inspired by the presentation of Giannopolous [Gia97] and Section 8.3 follows [Rot14].

The first natural question is: given the number of elements n and the number of sets m, what is the best upper bound on $\text{disc}(\mathcal{S})$ that one can guarantee without knowing any further properties of the set system. This question was resolved by Spencer [Spe85]:

Theorem 8.1 (Spencer '85)**.** *Let $\mathcal{S}$ be a set system with m sets and $n \leq m$ elements. Then disc$(\mathcal{S}) \leq O(\sqrt{n \cdot \log(\frac{2m}{n})})$.*

For example if $m = n$, then $\text{disc}(\mathcal{S}) \leq O(\sqrt{n})$. Spencer showed that the hidden constant can be bounded by 6, which explains the title "Six Standard Deviations Suffice" of Spencer's result. Before we come to Spencer's Theorem, let us see what one can get from simple measure concentration. We use the following well-known bound:

Theorem 8.2 (Chernov Bound)**.** *Let $X_1, \ldots, X_n$ independently distributed random variables with $|X_i| \leq 1$ and $\mathbb{E}[X_i] = 0$ for $i = 1, \ldots, n$. Then for any $\lambda \geq 0$ one has*

$$\Pr\left[\left|\sum_{i=1}^{n} X_i\right| > \lambda\sqrt{n}\right] \leq 2e^{-\lambda^2/2}.$$

Theorem 8.3. *For any set system $\mathcal{S}$ with m sets and maximum set size $s_{\max} := \max_{S\in\mathcal{S}} |S|$, one has disc$(\mathcal{S}) \leq O(\sqrt{s_{\max} \cdot \ln(m)})$.*

Proof. Let $\chi : [n] \to \{-1, 1\}$ be a *random coloring* that colors each element independently with a uniformly drawn color from $\{-1, 1\}$. Then for any set $S \in \mathcal{S}$ we know by the Chernov Bound (Theorem 8.2) that

$$\Pr\left[|\chi(S)| > \sqrt{|S| \cdot 2\ln(2m)}\right] \leq \frac{1}{2m}.$$

Then the union bound shows that

$$\Pr\left[\max_{S\in\mathcal{S}} |\chi(S)| > \sqrt{2s_{\max}\ln(2m)}\right] \leq \frac{1}{2}$$

and the claim follows. □

For example for $m = n$, this theorem only guarantees a bound of $O(\sqrt{n \cdot \log(n)})$, so we fall short by an extra $\sqrt{\log n}$ factor. To avoid this extra factor, we need to invest some work.

8.1 Finding partial colorings

If we look at colorings from an abstract point of view, then it is the question of when a symmetric convex set K must contain a point from $\{-1, 1\}^n$. One might wonder whether one could show a variant of Minkowski's Theorem that guarantees a point in $\{-1, 1\}^n$ as soon as K is large enough. For Minkowski's Theorem the "largeness" condition was just dependent on the volume. If we want an intersection with $\{-1, 1\}^n$, then we should only count the volume that is within a $\sqrt{n}$ distance to the origin. An elegant way to do this is by using the *Gaussian measure*. Recall that $N(0, 1)$ is the distribution over $x \in \mathbb{R}$ that has density $\frac{1}{\sqrt{2\pi}} e^{-x^2/2}$. This is the *standard normal distribution*, with $\mathbb{E}_{x\sim N(0,1)}[x] = 0$ and $\mathbb{E}_{x\sim N(0,1)}[x^2] = 1$. More generally, if we write $\boldsymbol{x} \sim N^n(0, 1)$ then this means that $\boldsymbol{x}$ is drawn from the *n-dimensional Gaussian distribution* that has density $\frac{1}{(2\pi)^{n/2}} \cdot e^{-\|\boldsymbol{x}\|_2^2/2}$. We see that this density is just the product of the coordinate densities, which means that a vector $\boldsymbol{x} \sim N^n(0, 1)$ can be generated by sampling the coordinates $x_1, \ldots, x_n \sim N(0, 1)$ independently. For a measurable set $K \subseteq \mathbb{R}^n$ we write $\gamma_n(K) := \Pr_{\boldsymbol{x}\sim N^n(0,1)}[\boldsymbol{x} \in K]$ as the *Gaussian measure* of that set. Before we come to the main technical part, we need two auxiliary lemmas. The first one gives a bound on the change of the Gaussian measure under shifting:

Lemma 8.4. *Let $K \subseteq \mathbb{R}^n$ be a symmetric, convex set, then $\gamma_n(\boldsymbol{u}+K) \geq e^{-\|\boldsymbol{u}\|_2^2/2} \cdot \gamma_n(K)$ for any $\boldsymbol{u} \in \mathbb{R}^n$.*

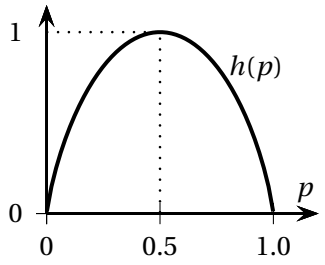
Proof. The set K is symmetric, so consider a pair $\{\boldsymbol{x}, -\boldsymbol{x}\} \subseteq K$ and how the density for that pair changes if we translate K (it was $\frac{1}{(2\pi)^{n/2}} e^{-\|\boldsymbol{x}\|_2^2/2}$ for both to begin with). The average ratio of their densities is then

$$\mathop{\mathbb{E}}_{\sigma\in\{-1,1\}}\left[\frac{\exp(-\frac{1}{2}\|\sigma\boldsymbol{x}+\boldsymbol{u}\|_2^2)}{\exp(-\frac{1}{2}\|\boldsymbol{x}\|_2^2)}\right] = \underbrace{\mathop{\mathbb{E}}_{\sigma\sim\{-1,1\}}[\exp(-\sigma\langle\boldsymbol{x},\boldsymbol{u}\rangle)]}_{\geq e^0=1} \cdot e^{-\|\boldsymbol{u}\|_2^2/2} \geq e^{-\|\boldsymbol{u}\|_2^2/2}.$$

Here we use Jensen's inequality and the convexity of the function $f(x) := e^{-\alpha\cdot x}$ for $\alpha \in \mathbb{R}$. □

Observe that for an individual point $\boldsymbol{x} \in K$ the drop in the density could exceed the factor of $e^{-\|\boldsymbol{u}\|_2^2/2}$ — we really used the symmetry of K here.

Another ingredient that we need are exponentially many points from the hypercube that have a large distance to each other. An elegant way of doing the counting without messy calculations is to rely on entropy. For a random variable X, the *entropy* is defined as $H(X) = \sum_{x\in\mathrm{range}(X)} \Pr[X=x] \cdot \log_2(\frac{1}{\Pr[X=x]})$, where range$(X)$ denotes all values that X can attain. One useful property is that entropy is *subadditive*, that means if $X = (X_1, \ldots, X_n)$ is a random vector, then $H(X) \leq \sum_{i=1}^n H(X_i)$. Moreover, for a binary random variable $X \in \{0,1\}$ with $\Pr[X=1] = p$, we have $H(X) = h(p)$ where $h(p) := p\log_2(\frac{1}{p}) + (1-p)\cdot\log_2(\frac{1}{1-p})$ is the *binary entropy function*.



Note that $H(X) \leq \log_2(|\mathrm{range}(X)|)$ and this is an equality if and only if X is the uniform distribution on range(X).

Lemma 8.5. *Let $Z := \{\boldsymbol{x} \in \{0,1\}^n \mid \|\boldsymbol{x}\|_1 \leq \frac{n}{10}\}$. Then $|Z| \leq 2^{n/2}$.*

Proof. Imagine we sample $\boldsymbol{x} \sim Z$ uniformly at random. Let $H(\boldsymbol{x})$ be the *entropy* of the random vector $\boldsymbol{x}$. Then

$$\log_2|Z| = H(\boldsymbol{x}) \overset{\text{subadditivity}}{\leq} \sum_{i=1}^n H(x_i) \leq \sum_{i=1}^n h(\underbrace{\Pr[x_i=1]}_{\in[0,\frac{1}{10}]}) \leq n \cdot \underbrace{h\left(\frac{1}{10}\right)}_{\leq 0.47} \leq 0.47n$$

Here we use that h is monotonically increasing on the interval $[0, \frac{1}{2}]$. Rearranging for $|Z|$ gives the claim. □

Now we can get a simple bound on points with large distance on the hypercube. Note that the particular constants are only chosen for convenience reasons.

Lemma 8.6. *There is a subset $X \subseteq \{-1,1\}^n$ with $|X| \geq 2^{n/2}$ and so that $\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_1 \geq \frac{n}{4}$ for all $\boldsymbol{x}, \boldsymbol{y} \in X$ with $\boldsymbol{x} \neq \boldsymbol{y}$.*

Proof. Imagine a graph $G = (\{-1,1\}^n, E)$ having an edge $\{\boldsymbol{x},\boldsymbol{y}\} \in E$ if $\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_1 \le \frac{n}{10}$. Then using Lemma 8.5, the maximum degree of that graph is at most $2^{n/2}$. Then picking greedily any maximal independent set in that graph will give $\frac{2^n}{2^{n/2}} = 2^{n/2}$ many points. □

Finally we can get an analogue of Minkowski's Theorem for hypercube points.

Lemma 8.7 (Partial Coloring Lemma)**.** *Let $K \subseteq \mathbb{R}^n$ be a symmetric convex set with $\gamma_n(K) \ge e^{-n/5}$. Then there is an $\boldsymbol{x} \in 2K \cap \{-1,0,1\}^n$ with $|supp(\boldsymbol{x})| \ge \frac{n}{10}$.*

Proof. We use Lemma 8.6 to obtain a set $X \subseteq \{-1,1\}^n$ with $|X| \ge 2^{n/2}$ and $\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_1 \ge \frac{n}{10}$ for any distinct pair $\boldsymbol{x},\boldsymbol{y} \in X$. Consider the translates $\frac{\boldsymbol{x}}{2} + K$ for $\boldsymbol{x} \in X$. Then

$$\sum_{\boldsymbol{x}\in X} \gamma_n\Big(\frac{\boldsymbol{x}}{2}+K\Big) \overset{\text{Lem. 8.4}}{\ge} \sum_{\boldsymbol{x}\in X} \underbrace{\gamma_n(K)}_{\ge e^{-n/5}} \cdot e^{-\|\boldsymbol{x}/2\|_2^2/2} \overset{|X|\ge 2^{n/2}}{\ge} 2^{n/2}\cdot e^{-n/5}\cdot e^{-n/8} > 1.$$

Since $\gamma_n(\mathbb{R}^n) = 1$, there must be distinct points $\boldsymbol{x}',\boldsymbol{x}'' \in X$ so that their translates overlap, say $\boldsymbol{z} \in (\frac{\boldsymbol{x}'}{2} + K) \cap (\frac{\boldsymbol{x}''}{2} + K)$. If we use again the Minkowski norm with respect to K, then this means that $\|\frac{\boldsymbol{x}'}{2} - \frac{\boldsymbol{x}''}{2}\|_K \le \|\frac{\boldsymbol{x}'}{2} - \boldsymbol{z}\|_K + \|\frac{\boldsymbol{x}''}{2} - \boldsymbol{z}\|_K \le 2$. Now, we define $\boldsymbol{x} := \frac{1}{2}(\boldsymbol{x}' - \boldsymbol{x}'') \in \{-1,0,1\}^n$ and $\|\boldsymbol{x}\|_K = \|\frac{\boldsymbol{x}'}{2} - \frac{\boldsymbol{x}''}{2}\|_K \le 2$. Moreover, $|\mathrm{supp}(\boldsymbol{x})| \ge \frac{n}{10}$ by the choice of X. □

In order to apply the Partial Coloring Lemma, we do need to be able show that the convex body K belonging to the set system has a large enough Gaussian measure. While calculating the measure directly should be quite painful, there is a very useful estimate that we can use. Here a *strip* is any set of the form $\{\boldsymbol{x} \in \mathbb{R}^n : |\langle \boldsymbol{x},\boldsymbol{u}\rangle| \le \lambda\}$ for $\boldsymbol{u} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ and $\lambda \ge 0$.

Lemma 8.8 (Šidák [Šid67], Khatri [Kha67])**.** *Let $K \subseteq \mathbb{R}^n$ be a symmetric convex body and $S \subseteq \mathbb{R}^n$ be a strip. Then $\gamma_n(K \cap S) \ge \gamma_n(K)\cdot\gamma_n(S)$.*

Proof. We will need the following fact (that we will not prove here):

Fact. The Gaussian measure γ_n is *log-concave*, that means

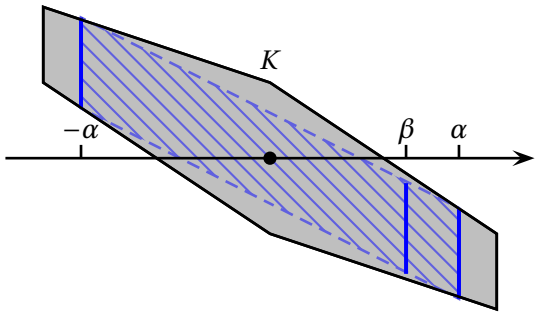
$$\gamma_n(\lambda A + (1-\lambda)B) \ge \gamma_n(A)^\lambda \cdot \gamma_n(B)^{1-\lambda}$$

for all compact sets $A, B \subseteq \mathbb{R}^n$ and all $0 \le \lambda \le 1$. Here $\lambda A + (1-\lambda)B := \{\lambda\boldsymbol{a} + (1-\lambda)\boldsymbol{b} \mid \boldsymbol{a} \in A,\ \boldsymbol{b} \in B\}$.

Suppose that $S = \{\boldsymbol{x} \in \mathbb{R}^n : |\langle \boldsymbol{x},\boldsymbol{u}\rangle| \le \lambda\}$ for $\boldsymbol{u} \in \mathbb{R}^n$ with $\|\boldsymbol{u}\|_2 = 1$ is the strip. It essentially suffices to show the following claim:

Claim. $F(\alpha) := \Pr[\boldsymbol{x} \in K \mid \langle \boldsymbol{x},\boldsymbol{u}\rangle = \alpha]$ is monotonically decreasing for $\alpha \ge 0$.
Proof. Since the Gaussian is rotationally symmetric, we may assume that $\boldsymbol{u} = \boldsymbol{e}_1$. Consider the $n-1$ dimensional slices $K_\alpha := \{(x_2,\dots,x_{n-1}) \mid (\alpha,x_2,\dots,x_n) \in K\}$ and let γ_{n-1} be the $n-1$ dimensional Gaussian measure. Consider $0 \le \beta \le \alpha$. For any point $(\alpha,\boldsymbol{x}) \in K$ one also has $(-\alpha,-\boldsymbol{x}) \in K$. We can write $\beta = \lambda\cdot\alpha + (1-\lambda)\cdot(-\alpha)$, then by convexity of K one has $K_\beta \subseteq \lambda K_\alpha + (1-\lambda)K_{-\alpha}$. By log-concavity of γ_{n-1} we obtain

$$\gamma_{n-1}(K_\beta) \ge \gamma_{n-1}(\lambda K_\alpha + (1-\lambda)K_{-\alpha}) \ge (\gamma_{n-1}(K_\alpha))^\lambda \cdot \underbrace{(\gamma_{n-1}(K_{-\alpha}))^{1-\lambda}}_{=\gamma_{n-1}(K_\alpha)} = \gamma_{n-1}(K_\alpha). \qquad \square$$



With the claim we then get

$$\gamma_n(K \cap S) = \Pr_{\alpha \sim N(0,1)}[|\alpha| \le \lambda] \cdot \mathop{\mathbb{E}}_{\alpha \sim N(0,1)}[F(\alpha) \mid |\alpha| \le \lambda] \ge \Pr_{\alpha \sim N(0,1)}[|\alpha| \le \lambda] \cdot \mathop{\mathbb{E}}_{\alpha \sim N(0,1)}[F(\alpha)] = \gamma_n(S) \cdot \gamma_n(K).$$

□

Recently it was proven that the Lemma even holds in the more general setting where also S is allowed to be any symmetric convex set.

8.2 Proof of Spencer's Theorem

Now we come to the actual proof of Spencer's Theorem. The main part consists in proving that there is a good partial coloring. We denote $\mathrm{supp}(\chi) := \{i \mid \chi(i) \neq 0\}$ as the *support*.

Lemma 8.9. *Let $\mathcal{S}$ be a set system with m sets and n elements with $n \le m$. Then there is a partial coloring $\chi : [n] \to \{-1,0,1\}$ with $|\chi(S)| \le O(\sqrt{n \log(\frac{2m}{n})})$ for all $S \in \mathcal{S}$ and $|\mathrm{supp}(\chi)| \ge \frac{n}{10}$.*

Proof. We consider the convex symmetric set

$$K := \left\{ \boldsymbol{x} \in \mathbb{R}^n : \left| \sum_{j \in S} x_j \right| \le \lambda \ \forall S \in \mathcal{S} \right\} = \bigcap_{S \in \mathcal{S}} \underbrace{\left\{ \boldsymbol{x} \in \mathbb{R}^n : \left| \sum_{j \in S} x_j \right| \le \lambda \right\}}_{=:\mathrm{strip}(S)}$$

with $\lambda := c\sqrt{n \log(\frac{2m}{n})}$ and $c > 0$ is a large enough universal constant. Then

$$\gamma_n(K) \overset{\text{Sidak-Kathri}}{\ge} \prod_{S \in \mathcal{S}} \gamma_n(\mathrm{strip}(S)) = \prod_{S \in \mathcal{S}} \Pr_{g \sim N(0,1)}\left[|g| \le \frac{\lambda}{\sqrt{|S|}} \right] \ge \Big(\underbrace{\Pr_{g \sim N(0,1)}\left[|g| \le c\sqrt{\log(\tfrac{2m}{n})} \right]}_{\ge 1 - \frac{n}{100m}} \Big)^m \overset{c \text{ large}}{\ge} e^{-n/5}.$$

By the Partial Coloring Lemma (Lemma 8.7) we conclude that there is a vector $\boldsymbol{x} \in 2K \cap \{-1,0,1\}^n$ with $|\mathrm{supp}(\boldsymbol{x})| \ge \frac{n}{10}$. This is the desired partial coloring of discrepancy $2c\sqrt{n \log(\frac{2m}{n})}$. □

We now show Spencer's Theorem. To keep the calculations a bit simpler, we only prove the balanced case with $m = n$.

Theorem 8.10 (Spencer's Theorem)**.** *Let $\mathcal{S}$ be a set system with n sets and n elements. Then $\mathrm{disc}(\mathcal{S}) \le O(\sqrt{n})$.*

Proof. We apply Lemma 8.9 to find a partial coloring. We color the elements as suggested and remove them from the set system. Then we repeat this procedure until all elements are colored eventually. Let $\chi^{(1)},\dots,\chi^{(T)} : [n] \to \{-1,0,1\}$ be the obtained partial colorings where $T \leq O(\log n)$. Note that at the beginning of iteration $t \in \{1,\dots,T\}$ the number of elements left is at most $n \cdot 0.9^{t-1}$, while the number of sets remains potentially flat at n. For the tth partial coloring, one has then

$$\operatorname{disc}(\chi^{(t)},\mathcal{S}) \overset{\text{Lem. 8.9}}{\leq} c\sqrt{n \cdot 0.9^{t-1} \cdot \log\Big(\frac{2n}{n \cdot 0.9^{t-1}}\Big)} \leq c\sqrt{n} \cdot 0.9^{(t-1)/2} \cdot t$$

Then $\chi := \sum_{t=1}^{T} \chi^{(t)} \in \{-1,1\}^n$ is the combined coloring with discrepancy

$$\operatorname{disc}(\chi,\mathcal{S}) \leq \sum_{t=1}^{T} \operatorname{disc}(\chi^{(t)},\mathcal{S}) \leq c\sqrt{n} \underbrace{\sum_{t=1}^{\infty} t \cdot 0.9^{(t-1)/2}}_{=O(1)} \leq O(\sqrt{n}).$$

Here we use that the sum is geometrically convergent. □

8.3 A constructive proof

Minkowski's Theorem and the Partial Coloring Lemma were both proven using the *pigeonhole principle* and a polynomial time algorithm realizing Minkowski's Theorem would break various lattice-based cryptosystem and hence is unlikely to exist. In contrast, for the partial coloring lemma, there are polynomial time algorithms. The first algorithm finding the coloring guaranteed by Spencer's Theorem was due to Bansal [Ban10] with an elegant generalization of Lovett and Meka [LM12]. Here we give the most general form due to [Rot14]:

Theorem 8.11. *Let $K \subseteq \mathbb{R}^n$ be symmetric and convex with $\gamma_n(K) \geq e^{-\varepsilon n}$. Then in polynomial time one can find a point $\boldsymbol{y} \in K \cap [-1,1]^n$ with $|\{i \mid y_i \in \{-1,1\}\}| \geq \varepsilon n$. Here $\varepsilon > 0$ is a small enough constant.*

Note that the partial coloring obtained from non-constructive arguments would be of the form $\boldsymbol{y} \in \{-1,0,1\}^n$ while here entries will indeed be from $[-1,1]$. But one can still use this Theorem to obtain Spencer's Theorem in a constructive way. The algorithm is surprisingly simple:

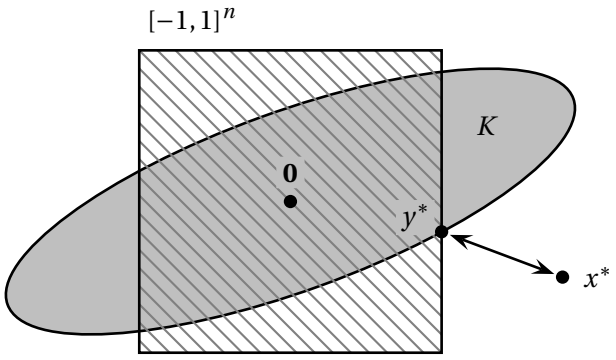
Algorithm:

(1) take a random Gaussian vector $\boldsymbol{x}^* \sim N^n(0,1)$

(2) compute the point $\boldsymbol{y}^* = \operatorname{argmin}\{\|\boldsymbol{x}^* - \boldsymbol{y}\|_2 \mid \boldsymbol{y} \in K \cap [-1,1]^n\}$

(3) return $\boldsymbol{y}^*$

The algorithm can be visualized as follows:



The algorithm solves a convex optimization problem, which can be done in polynomial time. By definition we will have $\boldsymbol{y}^* \in K \cap [-1,1]^n$. It only remains to prove that with high probability $\Omega(n)$ many coordinates in $\boldsymbol{y}^*$ will be from $\{-1,1\}$. For a compact set Q, let $d(\boldsymbol{x}, Q) := \min\{\|\boldsymbol{x} - \boldsymbol{y}\|_2 : \boldsymbol{y} \in Q\}$ be the Euclidean distance of $\boldsymbol{x}$ to Q. For the analysis, we will show a sequence of lemmas.

Lemma 8.12. *One has* $\Pr_{\boldsymbol{x}^* \sim N^n(0,1)}[d(\boldsymbol{x}^*, K \cap [-1,1]^n) \geq \frac{1}{5}\sqrt{n}] \geq 1 - 2^{-\Omega(n)}$.

Proof. We can argue that $\boldsymbol{x}^*$ has already a distance of $\Omega(\sqrt{n})$ to the hypercube $[-1,1]^n$. A simple calculation shows that $\Pr_{\boldsymbol{x}^* \sim N^n(0,1)}[|x_i| \geq 2] = 2\int_2^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt > \frac{1}{25}$. Then with probability $1 - e^{-\Omega(n)}$ we have $d(\boldsymbol{x}^*, [-1,1]^n) \geq \sqrt{\frac{n}{25} \cdot (2-1)^2} = \frac{1}{5} \cdot \sqrt{n}$. □

Let us denote $I^* := \{i \in [n] : y_i^* \in \{-1,1\}\}$ as the coordinates where $\boldsymbol{y}^*$ is on the boundary of the hypercube. Moreover, for $I \subseteq [n]$ let us define

$$K(I) := \{\boldsymbol{x} \in K : |x_i| \leq 1 \ \forall i \in I\}.$$

as the intersection of K with the coordinate strips for indices in I. In particular $K(\emptyset) = K$ and $K([n]) = K \cap [-1,1]^n$.

Lemma 8.13. *One always has* $d(\boldsymbol{x}^*, K \cap [-1,1]^n) = d(\boldsymbol{x}^*, K(I^*))$.

Proof. Note that we can write

$$d(\boldsymbol{x}^*, K) = \min\{\|\boldsymbol{x}^* - \boldsymbol{y}\|_2 : \boldsymbol{y} \in K \text{ and } |y_i| \leq 1 \ \forall i \in [n]\}. \qquad (*)$$

One can interpret the right hand side of $(*)$ as a *convex optimization problem*. We know that for indices $i \notin I^*$, the constraints $|y_i| \leq 1$ are not tight for the optimum solution $\boldsymbol{y}^*$. It is a well-known (and easy to prove) fact that dropping constraints in a convex optimization problem that were not tight for the optimum solution will not change the optimum value. Hence

$$(*) = \min\{\|\boldsymbol{x}^* - \boldsymbol{y}\|_2 : \boldsymbol{y} \in K \text{ and } |y_i| \leq 1 \ \forall i \in I^*\} = d(\boldsymbol{x}^*, K(I^*))$$

and the claim is proven. □

Lemma 8.14. *Fix any index set* $|I| \leq \varepsilon n$ *with* $\varepsilon > 0$ *small enough. Then*

$$\Pr_{\boldsymbol{x} \sim N^n(0,1)}\left[d(\boldsymbol{x}, K(I)) \geq \frac{1}{5}\sqrt{n}\right] \leq 2e^{-n/200}.$$

Proof. First we claim that the measure of $K(I)$ is still large. We can apply the Lemma of Šidák and Khatri (Lemma 8.8) to lower bound the measure of $K(I)$ as

$$\gamma_n(K(I)) \geq \gamma_n(K) \cdot \prod_{i \in I} \underbrace{\gamma_n(\{\boldsymbol{x} \in \mathbb{R}^n : |x_i| \leq 1\})}_{\geq e^{-1/2}} \geq \gamma_n(K) \cdot e^{-|I|/2} \overset{|I| \geq \varepsilon n}{\geq} e^{-\varepsilon n} \cdot e^{-(\varepsilon/2)n} \geq e^{-2\varepsilon n}$$

using that strips of width 2 have measure at least $e^{-1/2}$.

Next we want to argue that it is unlikely that a random Gaussian vector is far from a set that is as large as $K(I)$. For this we need the following concentration result. Recall that a function $F : \mathbb{R}^n \to \mathbb{R}$ is *Lipschitz* if $|F(\boldsymbol{x}) - F(\boldsymbol{y})| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

Lemma 8.15 (Concentration for Lipschitz Functions)**.** *Let $F : \mathbb{R}^n \to \mathbb{R}$ be an Lipschitz function with Gaussian mean $\mu := \mathbb{E}_{\boldsymbol{g} \sim N^n(0,1)}[F(\boldsymbol{g})]$. Then $\Pr_{\boldsymbol{g} \sim N^n(0,1)}[|F(\boldsymbol{g}) - \mu| \geq \lambda] \leq 2e^{-\lambda^2/2}$ for any $\lambda \geq 0$.*

A natural Lipschitz function is the distance function $F(\boldsymbol{x}) := d(\boldsymbol{x}, K(I))$. Then $\Pr_{\boldsymbol{x} \sim N^n(0,1)}[|d(\boldsymbol{x}, K(I)) - \mu| \geq \frac{1}{10}\sqrt{n}] \leq 2e^{-n/200}$. Let us choose ε small enough so that $\gamma_n(K(I)) > 2e^{-n/200}$, then we know that $\mu \leq \frac{1}{10}\sqrt{n}$ since $d(\boldsymbol{x}, K(I)) = 0$ for all $\boldsymbol{x} \in K(I)$. The claim follows. □

Lemma 8.16. *One has $\Pr[|I^*| \leq \varepsilon n] \leq 2^{-\Omega(n)}$, if $\varepsilon > 0$ is a small enough constant.*

Proof. We claim that there are two types of bad events:

(A) One has $d(\boldsymbol{x}^*, K \cap [-1,1]^n) \leq \frac{1}{5}\sqrt{n}$.

(B) For $I \subseteq [n]$ with $|I| \leq \varepsilon$ one has $d(\boldsymbol{x}^*, K(I)) \geq \frac{1}{5}\sqrt{n}$.

First we claim that if none of the events in (A) and (B) happen, then for all $I \subseteq [n]$ with $|I| \leq \varepsilon n$ one has

$$d(\boldsymbol{x}^*, K(I)) < \frac{1}{5}\sqrt{n} < d(\boldsymbol{x}^*, K \cap [-1,1]^n) = d(\boldsymbol{x}^*, K(I^*))$$

and hence $|I^*| > \varepsilon n$. So it remains to bound the probability of (A) or (B) happening. We can take the union bound over the bad events:

$$\underbrace{\Pr\left[d(\boldsymbol{x}^*, K \cap [-1,1]^n) \leq \frac{1}{5}\sqrt{n}\right]}_{\leq 2^{-\Omega(n)} \text{ by Lem 8.12}} + \underbrace{\sum_{|I| \leq \varepsilon n}}_{\leq 2 \cdot \binom{n}{\varepsilon n} \text{ terms}} \underbrace{\Pr\left[d(\boldsymbol{x}^*, K(I^*)) \geq \frac{1}{5}\sqrt{n}\right]}_{\leq 2e^{-n/200} \text{ by Lem. 8.14}} \leq 2^{-\Omega(n)}$$

if we choose $\varepsilon > 0$ small enough. This shows the claim. □

Bibliography

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 601–610, 2001.

[Ban93a] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

[Ban93b] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

[Ban10] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 3–10. IEEE Computer Society, 2010.

[Bel77] David E. Bell. A theorem concerning the integer lattice. *Studies in Applied Mathematics*, 56(2):187–188, 1977.

[CHKM92] W. J. Cook, M. Hartmann, R. Kannan, and C. McDiarmid. On integer points in polyhedra. *Combinatorica*, 12(1):27–37, 1992.

[Dad14] Daniel Dadush. A randomized sieving algorithm for approximate integer programming. *Algorithmica*, 70(2):208–244, October 2014.

[Doi73] J.P. Doignon. Convexity in cristallographical lattices. *Journal of Geometry*, 3:71–85, 1973.

[ES06] F. Eisenbrand and G. Shmonin. Carathéodory bounds for integer cones. *Oper. Res. Lett.*, 34(5):564–568, 2006.

[Gia97] Apostolos Giannopoulos. On some vector balancing problems. *Studia Mathematica*, 122(3):225–234, 1997.

[GR14] Michel X Goemans and Thomas Rothvoß. Polynomiality for bin packing with a constant number of item types. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 830–839. SIAM, 2014.

[Har88] M. Hartmann. Cutting planes and the complexity of the integer hull, 1988.

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 193–206, New York, NY, USA, 1983. ACM.

[Kan87a] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, August 1987.

[Kan87b] Ravindran Kannan. Algorithmic geometry of numbers. *Annual Review of Comp. Sci*, 2:231–267, 1987.

[Kha67] C. G. Khatri. On certain inequalities for normal distributions and their applications to simultaneous confidence bounds. *Ann. Math. Statist.*, 38:1853–1867, 1967.

[Len83] Jr. Lenstra, H. W. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):pp. 538–548, 1983.

[LM12] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 61–67. IEEE Computer Society, 2012.

[Mat99] J. Matousek. *Geometric Discrepancy: An Illustrated Guide*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 1999.

[Mat02] Jiri Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[MV10] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 351–358, 2010.

[MV13] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. *SIAM J. Comput.*, 42(3):1364–1391, 2013.

[Odl90] A. M. Odlyzko. The rise and fall of knapsack cryptosystems. *Cryptology and Computational Number Theory*, pages 75–88, 1990.

[Pei13] Chris Peikert. Lattices in cryptography, 2013.

[Reg09] Oded Regev. Lecture notes on lattices, 2009.

[Rot14] Thomas Rothvoß. Constructive discrepancy minimization for convex sets. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 140–145. IEEE Computer Society, 2014.

[Sca77] H. Scarf. An observation on the structure of production sets with indivisibilities. *Proceedings, National Academy of Sciences*, Vol. 74(9), 1977.

[Sch99] Alexander Schrijver. *Theory of linear and integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999.

[Šid67] Z. Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *J. Amer. Statist. Assoc.*, 62:626–633, 1967.

[Spe85] Joel Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289:679–706, 1985.