Utrecht University

Master Thesis

An application of LLL in geodesic continued fractions

Author:
Riaane Maes
3061787

Supervisor:
Prof. dr. Frits Beukers
Second Examiner:
Dr. Karma Dajani

June, 2015

Contents

1 Multidimensional continued fractions and small linear forms 5

1.1 One dimensional continued fractions 5

1.2 Multidimensional continued fractions 6

2 A theorem by Cusick and Krass 12

2.1 The value $c(\boldsymbol{\alpha})$ 12

2.2 The unit conjecture 18

3 Reduction of quadratic forms 21

3.1 Quadratic forms 21

3.2 Minkowski reduction 21

3.3 LLL-reduction 22

4 Geodesic continued fraction 25

4.1 Geodesic algorithm with LLL-reduction 26

5 Implementation and results 32

5.1 Short description of the code 32

5.2 Tests with random $\boldsymbol{\alpha}$ and varying dimension 33

5.3 Tests with number fields 40

References 45

A The mathematica notebook 47

Introduction and notation

Introduction

Given an irrational $\alpha \in \mathbb{R}$, we can use the continued fraction algorithm to approximate α by rationals. By Dirichlets theorem we know that there are infinitely many $(p, q) \in \mathbb{Z}$ such that $|\alpha - \frac{p}{q}| < 1/q^2$. When we extend this idea to higher dimensions, there are two dual cases to consider. Given $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ we can look for good simultaneous approximations of the α_i by fractions with the same denominator. So look for $(q, \boldsymbol{p}) \in \mathbb{Z}^{n+1}$ such that

$$\max_{i=1,\ldots,n} |\alpha_i - \frac{p_i}{q}| \text{ is small.}$$

In this thesis we will focus on the dual case, which is the problem of finding small values of linear forms in the α_i at integers points. So we will look for integers $(q, \boldsymbol{p}) \in \mathbb{Z}^{n+1}$ such that

$$|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \text{ is small.}$$

In Section 1 we describe the duality between these problems and we prove that there are infinitely many $(q, \boldsymbol{p}) \in \mathbb{Z}^{n+1}$ such that $|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot \max_i |p_i|^n < 1$. Most of the theory in this section comes from [1].

Suppose that $(1, \alpha_1, \ldots, \alpha_n)$ is an integral basis for a real number field F. Cusick and Krass [2] describe a lower bound for the smallest value c such that

$$|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot \max_i |p_i|^n < c$$

has infinitely many solutions. We will prove the theorem of Cusick and Krass in Section 2. For this we need the Unit Conjecture, which we also discuss in Section 2.

Next, we want an algorithm to actually compute these $(q, \boldsymbol{p}) \in \mathbb{Z}^{n+1}$. Brentjes [3] describes several algorithms for this purpose, for example the Jacobi-Perron algorithm and Brun's algorithm, but none of them are powerful in high dimensions. In 1850, Hermite proposed to use the quadratic form

$$Q_t = (x_1 - \alpha_1 y)^2 + \ldots, (x_n - \alpha_n y)^2 + ty^2$$

to find good simultaneous approximation with fractions. When the integers $(q, \boldsymbol{p})$ minimize Q_t this gives an approximation for which $\max_{i=1,\ldots,n} |q\alpha_i - p_i| < \gamma_n q^{-1/n}$ for a constant γ_n depending only on n. This idea was picked up by J.C. Lagarias [4] and in 1994 he proposed to use a geodesic algorithm which is based on Minkowski reduction of quadratic forms. He proposed to start with the quadratic form Q_t and let t decrease to 0. In the process he performed a change of variables so that the form remains Minkowski reduced. This will lead to a sequence of integers $(q, \boldsymbol{p})$ for which Q_t is minimal, and hence leads to approximations of the α_i. The advantage of Minkowski reduction is that the constrains that have to be met are linear in the parameter t and the number of constrains is finite. The disadvantage is that the number of constrains grows exponentially with the dimension. The LLL-reduction algorithm, named after its inventors Lenstra, Lenstra en Lovasz [5], is known to be powerful in high dimensions. Suggestions to use LLL-reduction instead of Minkowski reduction for this geodesic algorithm are made for example in [6]. The disadvantage of LLL-reduction is that the constrains are not linear in the parameter t. Now Beukers [7] observed that the LLL-conditions can be reformulated so that they are still linear in t, so this gives reason to implement and test this geodesic algorithm based on LLL-reduction. In Section 3 we describe Minkowski and LLL-reduction. The geodesic algorithm is described in Section 4.

For this thesis we implemented the geodesic algorithm based on LLL-reduction in `Mathematica` and performed tests to see how well the algorithm works. We performed tests with random $\boldsymbol{\alpha} \in \mathbb{R}^n$ to see how many changes of variables are needed to reach an approximation of certain quality. We did this up to dimension 25. We also run the algorithm with an integral bases of a real number fields of degree ≤ 6 with small discriminant. We compared the output of the algorithm with the lower bound given by Cusick and Krass and we used the algorithm to find integer elements of small norm in these number fields. The implementation and the results are described in Section 5.

Notation Let $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ and $(p_1, \ldots, p_n) \in \mathbb{Z}^n$. By $||p_1\alpha_1 + \ldots + p_n\alpha_n||$ we denote the distance of $p_1\alpha_1 + \ldots + p_n\alpha_n$ to the nearest integer. Sometimes we write $q + p_1\alpha_1 + \ldots + p_n\alpha_n$, then $q \in \mathbb{Z}$ is always chosen in such a way that $|q + p_1\alpha_1 + \ldots + p_n\alpha_n| = ||p_1\alpha_1 + \ldots + p_n\alpha_n||$. We write $||\boldsymbol{p}||_\infty = \max_{i=1,\ldots,n} |p_i|$ for the supremum norm and $||\boldsymbol{p}||_2 = \sqrt{p_1^2 + \ldots + p_n^2}$ for the Euclidean norm of $(p_1, \ldots, p_n)$.

Let $F = \mathbb{Q}(\alpha)$ be a real number field of degree $n+1$ where α is a root of an irreducible polynomial. Suppose $\mathbb{Q}(\alpha)$ has $r+1$ real and $2s$ complex embeddings, then we denote by $\alpha^{(j)}$ for $j = 0, 1, \ldots, r$ the real conjugates of α and by $\alpha^{(j)}$ for $j = r+1, \ldots, n$ the complex embeddings of α. Note that $\alpha^{(j)} = \overline{\alpha}^{(s+j)}$ for $j = r+1, \ldots, r+s$. Most of the time we just write α for $\alpha^{(0)}$. For each element $\beta \in F$ we define $N(\beta) = \prod_{j=0}^{n} \beta^{(j)}$ to be the norm of β. By O_F we denote the ring of integers of F and O_F^* is the unit group of F. Recall that $N(\beta) = \pm 1$ if and only if $\beta \in O_F^*$.

Acknowledgment I would like to thank Frits Beukers for the idea of this thesis, for his suggestions along the way and for taking the time to help me with all my questions.

1 Multidimensional continued fractions and small linear forms

1.1 One dimensional continued fractions

As an introduction to the theory of multidimensional continued fractions we will shortly describe the one dimensional case. The proofs of the statements made in this subsection can be found in Cassels [1]. Suppose $\alpha \in \mathbb{R}$, then we are interested in finding integers p, q such that $\frac{p}{q}$ lies close to α. The following theorem gives an idea of how well a real number can be approximated.

Theorem 1.1 (Dirichlet). *Let $\alpha \in \mathbb{R}$ and $Q > 1$. Then there exist integers q and p such that*

$$|q\alpha - p| \leq Q^{-1} \text{ and } 0 < q \leq Q.$$

Proof. This can be proven with the pigeon-hole principle. Look at all values $\{q\alpha\}$ for $q = 0, \ldots, Q$, where $\{\cdot\}$ denotes the fractional part of $q\alpha$. All these $Q+1$ values lie in the interval $[0, 1]$. There are Q intervals of the form $[\frac{n-1}{Q}, \frac{n}{Q})$ for $n = 1, \ldots, Q$ (where the interval is closed for $n = Q$), hence one of these intervals contains two of the values $\{q\alpha\}$. So there are integers q_1, q_2, r_1, r_2 such that $q_1\theta - r_1$ and $q_2\theta - r_2$ lie in the same interval of length Q^{-1}. Without loss of generality we can assume that $q_1 > q_2$. Now $|(q_1 - q_2)\alpha - (r_1 - r_2)| \leq Q^{-1}$ and $0 < q_1 - q_2 \leq Q$ as desired. □

Since this theorem holds for any $Q > 1$ it follows that there are infinitely many integers (p, q) such that $|q\alpha - p| < q^{-1}$. The continued fraction algorithm gives us a method to calculate these integers p and q. The algorithm runs as follows, where we write $[x]$ for the floor of x.

$$\begin{aligned}
&x_0 = \alpha \\
&a_0 = [x_0] \text{ and } x_1 = \frac{1}{x_0 - [x_0]} \\
&a_1 = [x_1] \text{ and } x_2 = \frac{1}{x_1 - [x_1]} \\
&\vdots \\
&a_n = [x_n] \text{ and } x_{n+1} = \frac{1}{x_n - [x_n]} \\
&\vdots
\end{aligned}$$

Now

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}}.$$

This algorithm terminates if and only if $\alpha \in \mathbb{Q}$. Suppose we cut of the algorithm at a_i, then we can write the right hand side as $\frac{p_i}{q_i}$ and we have

$$|\alpha - \frac{p_i}{q_i}| \leq \frac{1}{q_i^2},$$

hence we find infinitely many integers which fit the bound given by Dirichlet.

Example 1. Suppose $\alpha = \pi$, then we find $a_0 = 3, a_1 = 7, a_2 = 15, a_3 = 1$ which leads to the approximations 3, $\frac{22}{7}$, $\frac{333}{106}$ and $\frac{335}{113}$.

Now it seems natural to ask if this bound can be improved. So, given any $\alpha \in \mathbb{R}$, does there exist a constant $c > 0$ such that

$$|q\alpha - p| < cq^{-1}$$

has infinitely many solutions in integers p and q? It turns out that for almost all $\alpha \in \mathbb{R}$ and all $c > 0$ this equation has infinitely many solutions. Stated differently, the $\alpha \in \mathbb{R}$ for which a constant $c > 0$ exists such that the inequality $|q\alpha - p| < cq^{-1}$ does not have infinitely many solutions, have Lebesque measure zero. This is a special case of Theorem 1.8 which we will state later.

1.2 Multidimensional continued fractions

If we want to extend the theory above to higher dimensions, there are two dual cases to consider. The first one is the simultaneous approximation of a set of numbers $(\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n$ by fractions with the same denominator. Stated differently, we want to find $q \in \mathbb{Z}$ such that

$$\max_{1 \leq i \leq n} ||q\alpha_i|| \text{ is small.}$$

The dual case is where we want to find small values of linear forms in $(\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n$ with integer coefficients. That is, we want to find integers $\boldsymbol{p} \in \mathbb{Z}^n$ such that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \text{ is small.}$$

In this thesis we will focus on the latter case, but first we will prove the duality of these problems. For this, we need Minkowski's linear form theorem which is a direct consequence of Minkowski's convex body theorem.

Theorem 1.2 (Minkowski's Convex Body Theorem). *Let V be a symmetrical and convex subspace of $\mathbb{R}^n$. If*

$$vol(V) > 2^n,$$

then there exists a non-zero vector $\lambda \in V \cap \mathbb{Z}^n$.

Proof. Look at the subspace $\frac{1}{2}V = \{\boldsymbol{x} \in \mathbb{R}^n : 2\boldsymbol{x} \in V\}$. Then $vol(\frac{1}{2}V) = 2^{-n}vol(V) > 1$. For all integers $\boldsymbol{u} \in \mathbb{Z}^n$ we define the region ω_u to be the intersection of $\frac{1}{2}V$ and the hypercube $\{\boldsymbol{x} \in \mathbb{R}^n : u_i \leq x_i < u_i + 1\}$ (note that these hypercubes cover the whole of the $\mathbb{R}^n$). Now $vol(\bigcup_{\boldsymbol{u} \in \mathbb{Z}^n} \omega_{\boldsymbol{u}}) = vol(\frac{1}{2}V)$. Next, look at the regions $\omega'_{\boldsymbol{u}} = \{\boldsymbol{x} - \boldsymbol{u} : \boldsymbol{x} \in \omega_u\}$. All these regions lie in the hypercube $\{\boldsymbol{x} \in \mathbb{R}^n : 0 \leq x_i < 1\}$. Since $vol(\omega_{\boldsymbol{u}}) = vol(\omega'_{\boldsymbol{u}})$ we have that $vol(\bigcup_{\boldsymbol{u} \in \mathbb{Z}^n} \omega'_{\boldsymbol{u}}) > 1$ hence two of the $\omega'_{\boldsymbol{u}}$ must overlap. Suppose that this happens for $\omega'_{\boldsymbol{u}'}$ and $\omega'_{\boldsymbol{u}''}$. Then there are points $\boldsymbol{x}' \in \omega_{\boldsymbol{u}'}$ and $\boldsymbol{x}'' \in \omega_{\boldsymbol{u}''}$ such that $\boldsymbol{x}' - \boldsymbol{u}' = \boldsymbol{x}'' - \boldsymbol{u}''$. Now $\boldsymbol{x}', \boldsymbol{x}'' \in \frac{1}{2}V$ and $\lambda = \boldsymbol{x}' - \boldsymbol{x}'' \in \mathbb{Z}^n$. Since $\frac{1}{2}V$ is convex and symmetrical, also $\frac{1}{2}\lambda = \frac{1}{2}\boldsymbol{x}' - \frac{1}{2}\boldsymbol{x}'' \in \frac{1}{2}V$, hence $\lambda \in V$. □

Theorem 1.3 (Minkowski's linear form theorem). *The system of inequalities*

$$|\sum_{j=1}^{n} a_{1j}x_j| \leq c_1, \quad |\sum_{j=1}^{n} a_{ij}x_j| < c_i \quad (i = 2, \dots, n)$$

where $a_{ij}, c_i \in \mathbb{R}$ for $1 \leq j \leq i \leq n$, has a non-trivial integer solution $\boldsymbol{x} \in \mathbb{Z}^n$ provided that $c_1 \dots c_n \geq |\det(a_{ij})|$.

Proof. Define

$$V = \{\boldsymbol{x} \in \mathbb{R}^n : |\sum a_{1j}x_j| \le c_1 \text{ and } |\sum a_{ij}x_j| < c_i \text{ for } i = 2, \ldots, n\}.$$

Then V is bounded and $vol(V) = \frac{2^n c_1 \cdots c_n}{|\det(a_{ij})|}$. Suppose $c_1 \ldots c_n > |\det(a_{ij})|$, then the theorem follows immediately from Theorem 1.2. Now suppose $c_1 \ldots c_n = |\det(a_{ij})|$. Then for each $\epsilon > 0$ there exist an $\boldsymbol{x}^{(\epsilon)} \in \mathbb{Z}^n$ such that $|\sum a_{1j}x_j^{(\epsilon)}| < c_1 + \epsilon$ and $|\sum a_{ij}x_j^{(\epsilon)}| < c_i$. Since V is bounded, there are for each $\epsilon > 0$ only finitely many choices for $\boldsymbol{x}^{(\epsilon)}$. Let $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(\epsilon)}$, then $\boldsymbol{x}^{(0)}$ is the desired solution when $\epsilon \to 0$. □

Now we can prove the duality of the simultaneous approximation by fractions and the search for small values of linear forms in the $(\alpha_1, \ldots, \alpha_n)$.

Theorem 1.4. *Let $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ such that $1, \alpha_1, \ldots, \alpha_n$ are linearly independent over $\mathbb{Q}$.*

(i) Suppose that there exist integers $q_0, q_1, \ldots, q_n$ (with $q_0 \neq 0$) such that

$$\max_{i=1,\ldots,n} |\alpha_i q_0 - q_i| \le C \text{ and } |q_0| \le Q$$

for some constants C and Q with $0 < C < 1 \le Q$. Then there are integers $p_0, p_1, \ldots, p_n$ (not all zero) such that

$$|p_0 + p_1\alpha_1 + \ldots + p_n\alpha_n| \le D \text{ and } \max_{i=1,\ldots,n} |p_i| \le P,$$

where $D = nCQ^{1/n-1}$ and $P = nQ^{1/n}$.

(ii) Suppose that there exist integers $p_0, p_1, \ldots, p_n$ (not all zero) such that

$$|p_0 + p_1\alpha_1 + \ldots + p_n\alpha_n| \le D \text{ and } \max_{i=1,\ldots,n} |p_i| \le P$$

for some constants D and P with $0 < D < 1 \le P$. Then there are integers $q_0, q_1, \ldots, q_n$ (with $q_0 \neq 0$) such that

$$\max_{i=1,\ldots,n} |\alpha_i q_0 - q_i| \le C \text{ and } |q_0| \le Q,$$

where $C = nD^{1/n}$ and $Q = nPD^{1/n-1}$.

Proof. First we prove (i). We define two sets of $n+1$ forms in $n+1$ variables.

$$f_i(q_0, q_1, \ldots, q_n) = \begin{cases} C^{-1}(\alpha_i q_0 - q_i) & \text{for } i = 1, \ldots, n. \\ Q^{-1}q & \text{for } i = n+1. \end{cases}$$

$$g_i(p_0, p_1, \ldots, p_n) = \begin{cases} Cp_i & \text{for } i = 1, \ldots, n. \\ -Q(p_0 + p_1\alpha_1 + \ldots + p_n\alpha_n) & \text{for } i = n+1. \end{cases}$$

By assumption we know that there exist integers $\boldsymbol{q} \in \mathbb{Z}^{n+1}$ such that $|f_i(\boldsymbol{q})| \le 1$ for $i = 1, \ldots, n+1$. We fix this value for $\boldsymbol{q}$ and define $\lambda = \max_i |f_i(\boldsymbol{q})| = |f_l(\boldsymbol{q})|$. Since $1, \alpha_1, \ldots, \alpha_n$ are linearly independent over $\mathbb{Q}$ we know that $0 < \lambda \le 1$. Next note that

$$\begin{aligned}\sum_{i=1}^{n+1} f_i(\boldsymbol{q})g_i(\boldsymbol{p}) &= p_1(\alpha_1 q - q_1) + \ldots + p_n(\alpha_n q - q_n) - q_0(p_0 + p_1\alpha_1 + \ldots + p_n\alpha_n) \\ &= -(p_0q_0 + p_1q_1 + \ldots + p_nq_n),\end{aligned} \tag{1}$$

hence $\sum_{i=1}^{n+1} f_i(\boldsymbol{q})g_i(\boldsymbol{p}) \in \mathbb{Z}$.

Now we look at the $n+1$ inequalities in the variables $(p_0, p_1, \ldots, p_n)$

$$\begin{aligned} &|\sum_{i=1}^{n+1} f_i(\boldsymbol{q})g_i(\boldsymbol{p})| < 1 \\ &|g_i(\boldsymbol{p})| \leq Q^{1/n}C \quad (1 \leq i \leq n+1, i \neq l). \end{aligned}$$

The determinant of the forms on the left hand side is λQC^n and the product of the values on the right hand side is QC^n. Since $\lambda \leq 1$ we know by Minkowski's linear form theorem that this system of inequalities has a non-trivial solution. Let $\boldsymbol{p} \in \mathbb{Z}^{n+1}$ be a solution, then for all $i \neq l$ we have $|g_i(\boldsymbol{p})| \leq Q^{1/n}C$. Also $\sum_{i=1}^{n+1} f_i(\boldsymbol{q})g_i(\boldsymbol{p}) = 0$ and thus $\lambda g_l = f_l g_l = -\sum_{i \neq l} f_i g_i$. Hence $|g_l| \leq n\lambda CQ^{1/n}$. Thus for $i = 1, \ldots, n+1$ we have

$$\left.\begin{array}{l} |Cp_i| \\ |Q(p_0 + p_1\alpha_1 + \ldots + p_n\alpha_n)| \end{array}\right\} \leq n \cdot C \cdot Q^{1/n}.$$

Thus $\max_i |p_i| < nQ^{1/n}$ and $|p_0 + p_1\alpha_1 + \ldots + p_n\alpha_n| < nCQ^{1/n-1}$, which proves part (i).

For (ii) almost the same arguments hold. We define

$$f_i(p_0, p_1, \ldots, p_n) = \begin{cases} P^{-1}p_i & \text{for } i = 1, \ldots, n. \\ D^{-1}(p_0 + p_1\alpha_1 + \ldots + p_n\alpha_n) & \text{for } i = n+1. \end{cases}$$

$$g_i(q_0, q_1, \ldots, q_n) = \begin{cases} P(\alpha_i q_0 - q_i) & \text{for } i = 1, \ldots, n. \\ Dq_0 & \text{for } i = n+1. \end{cases}$$

Now we know by assumption that there is a non-trivial $\boldsymbol{p} \in \mathbb{Z}^{n+1}$ such that $|f_i(\boldsymbol{p})| \leq 1$. We fix this value for $\boldsymbol{p}$ and define $\mu = \max_i |f_i(\boldsymbol{p})| = |f_l(\boldsymbol{p})|$. Since $1, \alpha_1, \ldots, \alpha_n$ are linearly independent over $\mathbb{Q}$ we know that $0 < \mu \leq 1$. Again

$$\sum_{i=1}^{n} f_i(\boldsymbol{q})g_i(\boldsymbol{p}) = -(p_0 q + p_1 q_1 + \ldots + p_n q_n) \in \mathbb{Z}$$

and we look at the system of inequalities in the variables $(q_0, q_1, \ldots, q_n)$

$$\begin{aligned} &|\sum_{i=1}^{n} f_i(\boldsymbol{q})g_i(\boldsymbol{p})| < 1 \\ &|g_i(\boldsymbol{q})| \leq D^{1/n}P \quad (1 \leq i \leq n+1, i \neq l). \end{aligned}$$

One can easily check that the determinant of the equations on the left hand side equals μDP^n and the product of the values on the right hand side equals DP^n. Hence we can apply Minkowski's linear form theorem again and by the same arguments as above we find

$$\left.\begin{array}{l} |P(\alpha_i q_0 - q_i)| \\ |Dq_0| \end{array}\right\} \leq n \cdot P \cdot D^{1/n}$$

from which part (ii) follows. □

We state a corollary of this theorem for later purposes.

Corollary 1.5. *Let $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ such that $1, \alpha_1, \ldots, \alpha_n$ are linearly independent over $\mathbb{Q}$. There exists a constant $\gamma > 0$ such that*

$$\max_{i=1,\ldots,n} ||\alpha_i q|| q^{\frac{1}{n}} \geq \gamma$$

for all non-zero $q \in \mathbb{Z}$ if and only if there exist a $\delta > 0$ such that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n \geq \delta$$

for all non-zero $\boldsymbol{p} \in \mathbb{Z}^n$.

Proof. First we prove the necessary condition. Suppose

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n > \delta$$

for all non-zero $\boldsymbol{p} \in \mathbb{Z}^n$. Let D, P, C and Q be as in theorem 1.4 (i). Then

$$\begin{aligned} D \cdot P^n &\geq \delta \Rightarrow \\ n \cdot C \cdot Q^{\frac{1}{n}-1} \cdot Q &\geq \delta \Rightarrow \\ C \cdot Q^{\frac{1}{n}} &\geq \delta \cdot n^{-1}. \end{aligned}$$

This still holds when we pick $C = \max_i ||q\alpha_i||$ and $Q = |q|$ hence

$$\max_{i=1,\ldots,n} ||q\alpha_i|| \cdot q^{\frac{1}{n}} \geq \gamma \text{ where } \gamma = \delta \cdot n^{-1}.$$

For the sufficient condition we assume that $\max_{i=1,\ldots,n} ||q\alpha_i|| \cdot q^{\frac{1}{n}} \geq \gamma$ and let C, Q, P and D be as in Theorem 1.4 (ii). Then

$$\begin{aligned} C \cdot Q^{\frac{1}{n}} &\geq \gamma \Rightarrow \\ C^n \cdot Q &\geq \gamma^n \Rightarrow \\ n \cdot D \cdot P \cdot D^{(1-n)/n} &\geq \gamma^n \Rightarrow D \cdot P^n \geq n^{-n}\gamma^{n^2} \end{aligned}.$$

Since Theorem 1.4 also holds when $D = ||p_1\alpha_1 + \ldots + p_n\alpha_n||$ and $P = ||\boldsymbol{p}||_\infty$ we have that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n \geq \delta \text{ with } \delta = n^{-n}\gamma^{n^2}.$$

□

To extend Theorem 1.1 to higher dimensions we use the following theorem.

Theorem 1.6. *Let*

$$L_i = \sum_j \alpha_{ij} x_j \quad (1 \leq i \leq m, 1 \leq j \leq n)$$

be m linear forms in n variables. For every real $X > 1$ there exists an $\boldsymbol{x} \in \mathbb{Z}^n$ and a $\boldsymbol{y} \in \mathbb{Z}^m$ such that

$$\begin{aligned} &|L_i(\boldsymbol{x}) - y_i| < X^{-n/m} \text{ for } (1 \leq i \leq m) \\ &|x_j| \leq X \text{ for } (1 \leq j \leq n). \end{aligned}$$

Proof. The determinant of the inequalities on the left hand side is equal to 1 and the product of the elements on the right hand side is 1. Hence the result follows from Minkowski's linear form theorem. □

Corollary 1.7. *For any $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$, there are infinitely many integer solutions to*

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n < 1.$$

Proof. Suppose that $1, \alpha_1, \ldots, \alpha_n$ are linearly dependent over $\mathbb{Q}$, then this is trivial. Now we suppose that $1, \alpha_1, \ldots, \alpha_n$ are linearly independent over $\mathbb{Q}$. Take $m = 1$ and apply Theorem 1.6 to see that for any $P > 1$ there exist $(p_0, p_1, \ldots, p_n) \in \mathbb{Z}^{n+1}$ such that

$$|p_1\alpha_1 + \ldots + p_n\alpha_n - p_0| < P^{-n} \text{ where } ||\boldsymbol{p}||_\infty \leq P.$$

Then $||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot P^n < 1$ and also $||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n < 1$. Now pick $\tilde{P} > P$ such that $||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot \tilde{P}^n > 1$, then again by Theorem 1.6 there exists $(\tilde{p}_0, \tilde{p}_1, \ldots, \tilde{p}_n) \in \mathbb{Z}^{n+1}$ with $||\tilde{\boldsymbol{p}}||_\infty < \tilde{P}$ such that $||\tilde{p}_1\alpha_1 + \ldots + \tilde{p}_n\alpha_n|| \cdot \tilde{P}^n < 1$ hence $||\tilde{p}_1\alpha_1 + \ldots + \tilde{p}_n\alpha_n|| \cdot ||\tilde{\boldsymbol{p}}||_\infty^n < 1$. We can repeat this argument to find infinitely many $\boldsymbol{p} \in \mathbb{Z}^n$ such that $||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n < 1$. □

Remark. By taking $n = 1$ we find that there are infinitely many solutions to $\max_{i=1,\ldots,n} ||q\alpha_i|| < q^{-\frac{1}{n}}$.

As in the one-dimensional case we can ask whether there exist a constant $c > 0$ such that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n < c$$

has infinitely many integer solutions for a given $\boldsymbol{\alpha} \in \mathbb{R}^n$. In particular we are interested in finding the infimum of those c for which this inequality has infinitely many solutions. We call this value $c(\boldsymbol{\alpha})$. Thus

$$c(\boldsymbol{\alpha}) = \liminf_{||\boldsymbol{p}||_\infty \to \infty} ||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n.$$

The $\boldsymbol{\alpha} \in \mathbb{R}^n$ for which $c(\boldsymbol{\alpha}) > 0$ have Lebesgue measure zero, which follows from a theorem of Cassels [1], which we state without proof. Here we follow the notation of Cassels and we say that almost no elements $\boldsymbol{\alpha} \in \mathbb{R}^n$ have a certain property when the elements with that property have Lebesgue measure zero, and almost all $\boldsymbol{\alpha} \in \mathbb{R}^n$ have a certain property when almost no elements lack it.

Theorem 1.8. *Let $\psi(q)$ be a monotonely decreasing function of the integer values $q > 0$ with $0 \leq \psi(q) \leq \frac{1}{2}$. Then the set of inequalities*

$$||q\alpha_j|| \leq \psi(q) \quad (1 \leq j \leq n)$$

has infinitely many integer solutions $q > 0$ for almost no or for almost all sets of n numbers $(\alpha_1, \ldots, \alpha_n)$ according as

$$\sum (\psi(q))^n$$

converges or diverges.

Corollary 1.9. *For almost all n-tuples $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$, the set of inequalities*

$$\max_{i=1,\ldots,n} ||q\alpha_i|| \leq cq^{-1/n}$$

has infinitely many solutions for any $c > 0$.

Proof. The function $cq^{-1/n}$ is monotonely decreasing in integers q and is $\leq \frac{1}{2}$ for small enough c. The sum $\sum (cq^{-1/n})^n = c^n \sum q^{-1}$ diverges, hence the corollary follows directly from Theorem 1.8. □

The previous theorem and corollary is stated for the simultaneous approximation of $\boldsymbol{\alpha} \in \mathbb{R}^n$ with fractions. We will study the dual case, hence we want the following.

Proposition 1.10. *Let $c(\boldsymbol{\alpha})$ be as above. For almost all $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ we have*

$$c(\boldsymbol{\alpha}) = \liminf_{||\boldsymbol{p}||_\infty \to \infty} ||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n = 0.$$

Proof. This is immediate from the necessary condition of Corollary 1.5 in combination with Corollary 1.9 □

In the special case where $1, \alpha_1, \ldots, \alpha_n$ are linearly independent elements of an algebraic number field, we have that $c(\boldsymbol{\alpha}) > 0$, as the following theorem will show.

Theorem 1.11. *Let $\alpha_1, \ldots, \alpha_n$ be any n numbers in a real algebraic number field of degree $n+1$ such that $1, \alpha_1, \ldots, \alpha_n$ are linearly independent over $\mathbb{Q}$. Then there is a constant $\gamma > 0$ (depending only on $\alpha_1, \ldots, \alpha_n$) such that*

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n \geq \gamma$$

for all $\boldsymbol{p} \in \mathbb{Z}^n$.

Proof. First observe that there exists a $q \in \mathbb{Z}$ such that $|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \leq \frac{1}{2}$. Since the α_i are algebraic numbers, there exists an integer u such that $u\alpha_i$ are algebraic integers for all $i = 1, \ldots, n$. Now define $\eta = u(q + p_1\alpha_1 + \ldots + p_n\alpha_n)$. For all conjugates $\eta^{(j)}$ of η we have

$$\begin{aligned}
|\eta^{(j)}| &\leq |\eta| + |\eta^{(j)} - \eta| \\
&\leq \frac{1}{2}|u| + |up_1(\alpha_1 - \alpha^{(j)}) + \ldots + up_n(\alpha_n - \alpha_n^{(j)})| \\
&\leq \frac{1}{2}|u| + ||\boldsymbol{p}||_\infty \cdot |u\frac{p_1}{||\boldsymbol{p}||_\infty}(\alpha_1 - \alpha^{(j)}) + \ldots + u\frac{p_n}{||\boldsymbol{p}||_\infty}(\alpha_n - \alpha_n^{(j)})| \\
&\leq \frac{1}{2}|u| + ||\boldsymbol{p}||_\infty \cdot \max\{|u\xi_1(\alpha_1 - \alpha^{(j)}) + \ldots + u\xi_n(\alpha_n - \alpha_n^{(j)})| : ||\boldsymbol{\xi}||_\infty = 1\} \\
&\leq C||\boldsymbol{p}||_\infty \text{ for a constant } C \text{ which depends only on } \boldsymbol{\alpha}.
\end{aligned}$$

Now we combine this with the fact that $|N(\eta)| \geq 1$ to find

$$1 \leq |\eta| \prod_{j=1}^{n} |\eta^{(j)}| \leq |\eta| \cdot C^n \cdot ||\boldsymbol{p}||_\infty^n$$

hence

$$|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot ||\boldsymbol{p}||_\infty^n \geq u^{-1}C^{-n}$$

which proves the theorem with $\gamma = u^{-1}C^{-n}$. □

Remark. For a given $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ we define the dual constant $c'(\boldsymbol{\alpha})$ to be the minimum value of those c' such that

$$\max_i ||q\alpha_i|| \cdot q^{\frac{1}{n}} < c'$$

has infinitely many solutions, hence $c'(\boldsymbol{\alpha}) = \liminf\limits_{q \to \infty} \max_i ||q\alpha_i|| \cdot q^{\frac{1}{n}}$. Now let $C = \sup c(\boldsymbol{\alpha})$ and $C' = \sup c'(\boldsymbol{\alpha})$ where the suprema are taken over all $\boldsymbol{\alpha} \in \mathbb{R}^n$, then by a theorem of Davenport [8] we have that $C = C'$. In particular, we do not have that $c(\boldsymbol{\alpha}) = c'(\boldsymbol{\alpha})$. For a formula for $c'(\boldsymbol{\alpha})$ see [2].

In the next chapter we will state a theorem of Cusick and Krass [2] which gives a lower bound for the constant $c(\boldsymbol{\alpha})$.

2 A theorem by Cusick and Krass

2.1 The value $c(\boldsymbol{\alpha})$

In the previous section we have seen that when $1, \alpha_1, \ldots, \alpha_n$ are linearly independent elements of a real number field, then $c(\boldsymbol{\alpha}) > 0$. A theorem by Cusick and Krass gives us a lower bound for this constant and under assumption of the Unit Conjecture this theorem gives us the value of $c(\boldsymbol{\alpha})$. Before we state this theorem we need some conventions on notation. Let F be a real number field of degree $n+1$ with $r+1$ real and $2s$ complex embeddings and suppose that $(1, \alpha_1, \ldots, \alpha_n)$ is an integral basis for O_F. We define the $n \times n$ matrix

$$A = \begin{pmatrix} \alpha_1^{(1)} - \alpha_1 & \cdots & \alpha_n^{(1)} - \alpha_n \\ \vdots & \ddots & \vdots \\ \alpha_1^{(n)} - \alpha_1 & \cdots & \alpha_n^{(n)} - \alpha_n \end{pmatrix}.$$

For all $\boldsymbol{p} \in \mathbb{Z}^n$ for which $(\alpha_1^{(j)} - \alpha_1)p_1 + \ldots + (\alpha_n^{(j)} - \alpha_n)p_n \neq 0$ for all $1 \leq j \leq n$ we define the signature function

$$\sigma(A\boldsymbol{p}) =$$
$$(\operatorname{sgn}((\alpha_1^{(1)} - \alpha_1)p_1 + \ldots + (\alpha_n^{(1)} - \alpha_n)p_n), \ldots, \operatorname{sgn}((\alpha_1^{(r)} - \alpha_1)p_1 + \ldots + (\alpha_n^{(r)} - \alpha_n)p_n)),$$

where the sign is taken over all real conjugates of $(\alpha_1^{(j)} - \alpha_1)p_1 + \ldots + (\alpha_n^{(j)} - \alpha_n)p_n$. This $\sigma(A\boldsymbol{p})$ is of the form $(u_1, \ldots, u_r)$, where $u_i = \pm 1$ for all $i = 1, \ldots, r$. We write Σ the for set of all possible signatures. Define

$$N_\sigma = \min\{|N(q + p_1\alpha_1 + \ldots + p_n\alpha_n)| : \sigma(A\boldsymbol{p}) = \sigma\},$$

thus N_σ is the minimum norm of all elements with signature N_σ. For a vector $(\nu_1, \ldots, \nu_n) \in \mathbb{R}^n$ we define

$$\Pi(A\boldsymbol{\nu}) = \prod_{j=1}^{n} (\alpha_1^{(j)} - \alpha_1)\nu_1 + \ldots + (\alpha_n^{(j)} - \alpha_n)\nu_n.$$

At last, we write

$$\mathbb{C}_+^{r,s} =$$
$$\{(x_1, \ldots, x_n) \in \mathbb{R}^r \times \mathbb{C}^{2s} : x_j \in \mathbb{R}_{>0} \text{ for } 1 \leq j \leq r,\ x_j = \overline{x}_{s+j} \text{ for } r+1 \leq j \leq r+s\}.$$

Conjecture 1 (Unit Conjecture). Let F be any real number field of degree $n+1$. For each $\boldsymbol{x} \in \mathbb{C}_+^{r,s}$ and for any $\epsilon > 0$ there exists a unit $\eta \in O_F$ with $\eta^{(j)} > 0$ for all $1 \leq j \leq r$, such that

$$\frac{\eta^{(j)}}{\eta^{(1)}} = \frac{x_j}{x_1} + \epsilon_j, \text{ with } |\epsilon_j| < \epsilon, \text{ for } j = 2, \ldots, n.$$

Theorem 2.1. *Let $(1, \alpha_1, \ldots, \alpha_n)$ be a basis for a real number field of degree $n+1$ and let*

$$c(\boldsymbol{\alpha}) = \liminf_{||\boldsymbol{p}||_\infty \to \infty} ||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n.$$

Then

$$c(\boldsymbol{\alpha}) \geq \min_{\sigma \in \Sigma} \frac{N_\sigma}{\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_\infty = 1,\ \sigma(A\boldsymbol{\nu}) = \sigma\}}.$$

Under the assumption of the Unit Conjecture we have equality.

Proof. For each $P > 1$ and $\sigma \in \Sigma'$ we define the set

$$\Omega_{P,\sigma'} := \{(p_1, \dots, p_n) \in \mathbb{Z}^n : ||p_1\alpha_1 + \dots + p_n\alpha_n|| < P^{-n}, ||\boldsymbol{p}||_\infty \leq P \text{ and } \sigma(A\boldsymbol{p}) = \sigma'\}.$$

Note that all of these sets are finite. By Theorem 1.6 we know that for each $P > 1$ we have that $\Omega_{P,\sigma'}$ is non-empty for at least one $\sigma' \in \Sigma$. Now for any $\epsilon > 0$ there exists a $Q > 1$ such that for all $P > Q$ we have that $||p_1\alpha_1 + \dots + p_n\alpha_n|| < \epsilon$ for all $\boldsymbol{p} \in \Omega_{P,\sigma'}$, for any $\sigma' \in \Sigma$. We write $\boldsymbol{p} \cdot \boldsymbol{\alpha} = p_1\alpha_1 + \dots + p_n\alpha_n$ and $\boldsymbol{p} \cdot \boldsymbol{\alpha^{(j)}} = p_1\alpha_1^{(j)} + \dots + p_n\alpha_n^{(j)}$. Now for all $\boldsymbol{p} \in \Omega_{P,\sigma'}$ with $P > Q$ we have

$$|q + \boldsymbol{p} \cdot \boldsymbol{\alpha^{(j)}}| - \epsilon \leq |\boldsymbol{p} \cdot \boldsymbol{\alpha^{(j)}} - \boldsymbol{p} \cdot \boldsymbol{\alpha}| \leq |q + \boldsymbol{p} \cdot \boldsymbol{\alpha^{(j)}}| + \epsilon \text{ for } j = 1, \dots, n.$$

Then

$$|N(q + \boldsymbol{p} \cdot \boldsymbol{\alpha})| = \prod_{j=0}^{n} |q + \boldsymbol{p} \cdot \boldsymbol{\alpha^{(j)}}|$$

$$\leq ||\boldsymbol{p} \cdot \boldsymbol{\alpha}|| \prod_{j=1}^{n} (|\boldsymbol{p} \cdot \boldsymbol{\alpha^{(j)}} - \boldsymbol{p} \cdot \boldsymbol{\alpha}| + \epsilon)$$

$$= ||\boldsymbol{p} \cdot \boldsymbol{\alpha}|| \prod_{j=1}^{n} (|(\alpha_1^{(j)} - \alpha_1)p_1 + \dots + (\alpha_n^{(j)} - \alpha_n)p_n| + \epsilon)$$

$$= ||\boldsymbol{p} \cdot \boldsymbol{\alpha}|| \cdot ||\boldsymbol{p}||_\infty^n \prod_{j=1}^{n} (|(\alpha_1^{(j)} - \alpha_1)\frac{p_1}{||\boldsymbol{p}||_\infty} + \dots + (\alpha_n^{(j)} - \alpha_n)\frac{p_n}{||\boldsymbol{p}||_\infty}| + \frac{\epsilon}{||\boldsymbol{p}||_\infty})$$

$$\leq ||\boldsymbol{p} \cdot \boldsymbol{\alpha}|| \cdot ||\boldsymbol{p}||_\infty^n \max\{\prod_{j=1}^{n} |(\alpha_1^{(j)} - \alpha_1)\nu_1 + \dots + (\alpha_n^{(j)} - \alpha_n)\nu_n + \epsilon'| : ||\boldsymbol{\nu}||_\infty = 1, \sigma(A\boldsymbol{\nu}) = \sigma'\}$$

where $\epsilon' = \frac{\epsilon}{||\boldsymbol{p}||_\infty}$. Then

$$||\boldsymbol{p} \cdot \boldsymbol{\alpha}|| \cdot ||\boldsymbol{p}||_\infty^n \geq \frac{|N(q + \boldsymbol{p} \cdot \boldsymbol{\alpha})|}{\max\{\prod_{j=1}^{n} |(\alpha_1^{(j)} - \alpha_1)\nu_1 + \dots + (\alpha_n^{(j)} - \alpha_n)\nu_n + \epsilon'| : ||\boldsymbol{\nu}||_\infty = 1, \sigma(A\boldsymbol{\nu}) = \sigma'\}}$$

thus

$$\liminf_{||\boldsymbol{p}||_\infty \to \infty} ||\boldsymbol{p} \cdot \boldsymbol{\alpha}|| \cdot ||\boldsymbol{p}||_\infty^n \geq \min_{\sigma \in \Sigma}\{\frac{N_\sigma}{\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_\infty = 1, \ \sigma(A\boldsymbol{\nu}) = \sigma\}}\}.$$

This proves the first part of the theorem. □

To prove the second part, we need to show that for each $\sigma \in \Sigma$ there are infinitely many elements in the optimal direction, which is stated in the following proposition.

Proposition 2.2. *Suppose that the maximum of*

$$\prod_{j=1}^{n} |(\alpha_1^{(j)} - \alpha_1)\nu_1 + \dots + (\alpha_n^{(j)} - \alpha_n)\nu_n|$$

is reached for $\boldsymbol{\xi} = (\xi_1, \dots, \xi_n)$ with $\sigma(A\boldsymbol{\xi}) = \sigma$. Then for each $\epsilon > 0$ there exists an element $\eta = q + p_1\alpha_1 + \dots + p_n\alpha_n \in O_F$ such that

1. $\left|\frac{p_j}{p_1} - \frac{\xi_j}{\xi_1}\right| < \epsilon$ *for $j = 2, \dots, n$.*

2. $|N(\eta)| = N_\sigma$.

Before we prove this, we look at the specific case where $\sigma = (1, \ldots, 1)$ and we prove that there is a unit $\eta \in O_F^*$ with these properties. This is done under the assumption of the unit conjecture.

Lemma 2.3. *Suppose $(\xi_1, \ldots, \xi_n) \in \mathbb{R}^n$ and $\sigma(A\boldsymbol{\xi}) = (1, \ldots, 1)$. Then for any $\epsilon > 0$ there exists a unit $\eta = q + p_1\alpha_1 + \ldots + p_n\alpha_n$ such that*

$$\left|\frac{p_j}{p_1} - \frac{\xi_j}{\xi_1}\right| < \epsilon \textit{ for } j = 2, \ldots, n.$$

Proof. Define

$$a_j = \frac{(\alpha_1^{(j)} - \alpha_1)\xi_1 + \ldots + (\alpha_n^{(j)} - \alpha_n)\xi_n}{(\alpha_1^{(1)} - \alpha_1)\xi_1 + \ldots + (\alpha_n^{(1)} - \alpha_n)\xi_n} \text{ for } j = 2, \ldots, n. \tag{2}$$

Since $\sigma(A\boldsymbol{\xi}) = (1, \ldots, 1)$ by assumption we have $a_j > 0$ for $j = 2, \ldots, r$ and by the definition of a_j we have $a_j = \overline{a}_{j+s}$ for $j = r+1, \ldots, r+s$. Now the conditions of the Unit Conjecture are met, hence for any $\epsilon > 0$ there exists an η such that $\eta^{(j)} > 0$ for $j = 1, \ldots, r$ and

$$\left|\frac{\eta^{(j)}}{\eta^{(1)}} - a_j\right| < \epsilon \text{ for } j = 2, \ldots, n. \tag{3}$$

We need the following claim, which we prove later.

Claim 2.4. For each $\epsilon > 0$ and $R > 1$ there exists an $\eta \in O_F^*$ such that $|\eta^{(1)}| > R$ and $\left|\frac{\eta^{(j)}}{\eta^{(1)}} - a_j\right| < \epsilon$ for $j = 2, \ldots, n$, where a_j is as defined above.

From this claim it follows that for each $R > 1$ there exists an $\eta \in O_F^*$ such that

$$\begin{aligned} &|\eta^{(1)}| > R \\ &|\eta^{(j)}| > R(|a_j| - \epsilon) \text{ for } j = 2, \ldots, n. \end{aligned}$$

Since

$$1 = |N(\eta)| = |\eta| \prod_{j=1}^{n} |\eta^{(j)}|$$

we have

$$|\eta| = \frac{1}{\prod_{j=1}^n |\eta^{(j)}|} < \frac{1}{R^n \prod_{j=2}^n (|a_j| - \epsilon)} < \epsilon',$$

where the right hand side can get less than any $\epsilon' > 0$ by taking R large and ϵ small enough. Next, note that we can write,

$$(\alpha_1^{(j)} - \alpha_1)p_1 + \ldots + (\alpha_n^{(j)} - \alpha_n)p_n = \eta^{(j)} - \eta \text{ for } j = 1, \ldots, n.$$

We combine the above to find

$$\begin{aligned} &\left|\frac{\eta^{(j)} - \eta}{\eta^{(1)} - \eta} - \frac{\eta^{(j)}}{\eta^{(1)}}\right| = \left|\eta \cdot \frac{\eta^{(j)} - \eta^{(1)}}{\eta^{(1)}(\eta^{(1)} - \eta)}\right| = \\ |\eta| \cdot &\left|\frac{\frac{\eta^{(j)}}{\eta^{(1)}} - a_j + a_j - 1}{\eta^{(1)} - \eta}\right| \leq \epsilon' \cdot \frac{\epsilon + |a_j - 1|}{R - \epsilon'} < \epsilon''. \end{aligned} \tag{4}$$

Again, ϵ'' can get arbitrarily small by taking R large and ϵ small enough. Combining (3) and (4) gives

$$\left|\frac{\eta^{(j)}-\eta}{\eta^{(1)}-\eta}-a_j\right|<\epsilon''' \text{ for } j=2,\ldots,n$$

where $\epsilon'''=\epsilon+\epsilon''$. Hence for $j=2,\ldots,n$ we have

$$\left|\frac{(\alpha_1^{(j)}-\alpha_1)p_1+\ldots+(\alpha_n^{(j)}-\alpha_n)p_n}{(\alpha_1^{(1)}-\alpha_1)p_1+\ldots+(\alpha_n^{(1)}-\alpha_n)p_n}-a_j\right|<\epsilon'''. \tag{5}$$

Now let $G:\mathbb{R}^{n-1}\to\mathbb{R}^{n-1}$ be the function

$$x_1,\ldots,x_{n-1}\mapsto(G_2,\ldots,G_n) \text{ where}$$

$$G_j=\frac{(\alpha_1^{(j)}-\alpha_1)+(\alpha_2^{(j)}-\alpha_2)x_1+\ldots+(\alpha_n^{(j)}-\alpha_n)x_{n-1}}{(\alpha_1^{(1)}-\alpha_1)+(\alpha_2^{(1)}-\alpha_2)x_1+\ldots+(\alpha_n^{(1)}-\alpha_n)x_{n-1}}.$$

Then (5) implies that

$$|G_j(\frac{p_2}{p_1},\ldots,\frac{p_n}{p_1})-G_j(\frac{\xi_2}{\xi_1},\ldots,\frac{\xi_n}{\xi_1})|<\epsilon''' \text{ for } j=2,\ldots,n. \tag{6}$$

To finish the proof we need the following claim, which we prove later.

Claim 2.5. Let the function $G:\mathbb{R}^{n-1}\to\mathbb{R}^{n-1}$ be as above. Than G is invertible around $(\frac{\xi_2}{\xi_1},\ldots,\frac{\xi_n}{\xi_1})$.

This claim, in combination with (6), gives that there exists a $\delta>0$ such that

$$\left|\frac{\xi_j}{\xi_1}-\frac{p_j}{p_1}\right|<\delta. \text{ for } j=2,\ldots,n.$$

This proves the lemma. □

Now we are ready to prove Proposition 2.2.

Proof of proposition 2.2. Suppose that the minimum of

$$\min_{\sigma\in\Sigma}\{\frac{N_\sigma}{\max\{\Pi(A\boldsymbol{\nu}):||\boldsymbol{\nu}||_\infty=1 \text{ and } \sigma(A\boldsymbol{\nu})=\sigma\}}\}$$

is taken by $(\xi_1,\ldots,\xi_n)\in\mathbb{R}^n$ with $\sigma(A\boldsymbol{\xi})=\sigma$. Now let $\theta\in O_F$ be such an element that $|N(\theta)|=N_\sigma$. Now we only need to multiply this θ with a unit in the right direction. That is, take a unit $\eta=q+p_1\alpha_1+\ldots+p_n\alpha_n$ such that

$$\theta\eta=\tilde{q}+\tilde{p_1}\alpha_1+\ldots+\tilde{p_n}\alpha_n \text{ with}$$
$$\left|\frac{\xi_j}{\xi_1}-\frac{\tilde{p}_j}{\tilde{p}_1}\right|<\epsilon.$$

This is possible since p_j/p_1 can have any ratio by the previous lemma. Now $|N(\theta\eta)|=|N(\theta)|\cdot|N(\eta)|=|N(\theta)|$ and since $\sigma(A\eta)=(1,\ldots,1)$, we have $\sigma(A\theta\eta)=\sigma(A\theta)$. □

To finish the proof of Theorem 2.1 we need to prove the Claim 2.4 and 2.5. For Claim 2.4 we need the following lemma.

Lemma 2.6. *For fixed $0 < \epsilon < 1$ and $R > 1$, there are only finitely many units $\psi \in O_F^*$ such that*

$$1 - \epsilon < \left|\frac{\psi^{(j)}}{\psi^{(1)}}\right| < 1 + \epsilon \text{ and } \frac{1}{R} < |\psi^{(1)}| < R.$$

Proof. Suppose $\psi \in O_F^*$ is such that

$$1 - \epsilon < \left|\frac{\psi^{(j)}}{\psi^{(1)}}\right| < 1 + \epsilon \text{ and } \frac{1}{R} < |\psi^{(1)}| < R.$$

Then

$$\frac{1}{R}(1-\epsilon) < |\psi^{(j)}| < (1+\epsilon)R \text{ for } j = 1, \dots, n.$$

Since ψ is a unit, we have $|\psi| = \frac{1}{\prod_{j=1}^n |\psi^{(j)}|}$ and the above gives

$$\frac{1}{R^n}\frac{1}{(1+\epsilon)^{n-1}} < |\psi| < R^n \frac{1}{(1-\epsilon)^{n-1}}.$$

Since the image of the map

$$\begin{aligned} &\Phi : O_F^* \to \mathbb{R}^{r+s+1} \\ &\psi \mapsto (\log|\psi|, \log|\psi^{(1)}|, \dots, \log|\psi^{(r)}|, \log|\psi^{(r+1)}| \dots, \log|\psi^{(r+s)}|) \end{aligned}$$

is a lattice in the hyperplane of $\mathbb{R}^{r+s+1}$ where $\sum x_i = 0$, we can conclude that there are only finitely many elements $\psi \in O_F^*$ for which $|\psi^{(j)}|$ is bounded for all $j = 0, \dots, n$. □

Proof of Claim 2.4. By combining the previous lemma with the Unit Conjecture, we know that given any $R > 1$ and small enough $\epsilon > 0$ there exists a unit $\psi \in O_F^*$ such that

$$1 - \epsilon < \left|\frac{\psi^{(j)}}{\psi^{(1)}}\right| < 1 + \epsilon \text{ and } |\psi^{(1)}| < \frac{1}{R} \text{ or } |\psi^{(1)}| > R.$$

Without loss of generality we can assume that $|\psi^{(1)}| > R$, otherwise simply replace ψ for $\frac{1}{\psi}$. Now let a_j be as in (2), then the unit conjecture gives us that for any $\tilde{\epsilon} > 0$ there exists a $\tilde{\psi} \in F^*$ such that

$$a_j - \tilde{\epsilon} < \left|\frac{\tilde{\psi}^{(j)}}{\tilde{\psi}^{(1)}}\right| < a_j + \tilde{\epsilon} \text{ for } j = 2, \dots, n.$$

Now let $\eta = \psi \cdot \tilde{\psi}$. Then for $j = 2, \dots, n$ we have

$$(1-\epsilon)(a_j - \tilde{\epsilon}) < \left|\frac{\eta^{(j)}}{\eta^{(1)}}\right| < (1+\epsilon)(a_j + \tilde{\epsilon})$$

hence $|\frac{\eta^{(j)}}{\eta^{(1)}} - a_j|$ can get arbitrarily small by taking ϵ and $\tilde{\epsilon}$ small enough. Also $|\eta^{(1)}| > |\tilde{\psi}^{(1)}|R$ and since this holds for any $R > 1$ this proves the claim. □

Proof of Claim 2.5. From the inverse function theorem it follows that G is invertible around $(\frac{\xi_2}{\xi_1}, \dots, \frac{\xi_n}{\xi_1})$ if $\det(J_G(\frac{\boldsymbol{\xi}_j}{\boldsymbol{\xi}_1})) \neq 0$, where J_G is the Jacobian matrix of G. For abbreviation we write $a_{ij} = \alpha_i^{(j)} - \alpha_i$ and $G_j = \Gamma_j / \Gamma_1$ where

$$\Gamma_i = (\alpha_1^{(i)} - \alpha_1) + (\alpha_2^{(i)} - \alpha_2)x_1 + \ldots + (\alpha_n^{(i)} - \alpha_n)x_{n-1}.$$

Then

$$J_G = \left(\frac{\Gamma_1 \cdot a_{ij} - \Gamma_j a_{i1}}{\Gamma_1^2}\right)_{1\le i,j\le n}.$$

By applying the Guassian row elimination $R_i \to R_i - \Gamma_i R_1$ for $i = 2, \ldots n$ we can show that the determinant of the matrix

$$M = \begin{pmatrix} 1 & a_{11}\cdot\Gamma_1^{-2} & \ldots & a_{n1}\Gamma_1^{-2} \\ \Gamma_2 & a_{12}\cdot\Gamma_1^{-1} & \ldots & a_{n2}\cdot\Gamma_1^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_n & a_{1n}\cdot\Gamma_1^{-1} & \cdots & a_{nn}\cdot\Gamma_1^{-1} \end{pmatrix}.$$

is equal to $\det(J_G)$ (here R_i denotes the i-th row of matrix M). Then

$$\det(J_G) = \frac{1}{\Gamma_1^{n+1}} \begin{pmatrix} \Gamma_1 & a_{11}\cdot\Gamma_1^{-1} & \ldots & a_{n1}\Gamma_1^{-1} \\ \Gamma_2\cdot\Gamma_1 & a_{12} & \ldots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_n\cdot\Gamma_1 & a_{1n} & \cdots & a_{nn} \end{pmatrix}.$$

By using Laplace expansion along the first column, one can show that

$$\Gamma_1^{n+1}\cdot\det(J_G) = \begin{pmatrix} a_{11} & a_{21} & \ldots & a_{n1} \\ a_{12} & a_{22} & \ldots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix} + \begin{pmatrix} \Gamma_1 - a_{11} & a_{21} & \ldots & a_{n1} \\ \Gamma_2 - a_{12} & a_{22} & \ldots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_n - a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}.$$

For the second matrix, we have that the first column is a linear combination of the other columns, hence its determinant is zero. By applying the Guassian row elimination $R_i \to R_i - R_1$ for $i = 2, \ldots n$ we can show that the determinant of first matrix equals

$$\begin{vmatrix} 1 & \alpha_1 & \cdots & \alpha_n \\ 1 & \alpha_1^{(1)} & \cdots & \alpha_n^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_1^{(n)} & \cdots & \alpha_n^{(n)} \end{vmatrix} = \sqrt{\Delta(F)},$$

where $\Delta(F)$ denotes the discriminant of F. Thus

$$\det(J_G) = \frac{1}{\Gamma_1^{n+1}}\cdot\sqrt{\Delta(F)}.$$

By the choice of $(\xi_1, \ldots, \xi_n)$ we know that $\Gamma_1^{n+1} \neq 0$ hence $\det(J_G(\xi_1, \ldots, \xi_n))$ is well-defined and unequal to zero, which proves the claim. □

Example 2. Let $F = \mathbb{Q}(\alpha)$ where α is a real root of $f(x) = x^3 + x^2 - 1$. Then $(1, \alpha, \alpha^2)$ is a basis for F. This field has 1 real and 2 complex embeddings, so $r = 0$ and $s = 1$. Then $\Sigma = \emptyset$ and $N_\sigma = 1$. Thus

$$c(\boldsymbol{\alpha}) = \frac{1}{\max\{\prod_{j=1}^{2} |(\alpha_1^{(j)} - \alpha_1)\nu_1 + \ldots + (\alpha_2^{(j)} - \alpha_2)\nu_2| : ||\boldsymbol{\nu}||_\infty = 1\}},$$

where $\alpha_1 = \alpha$ and $\alpha_2 = \alpha^2$. In Figure 1 we see a contour plot of

$$\prod_{j=1}^{2} |(\alpha_1^{(j)} - \alpha_1)\nu_1 + \ldots + (\alpha_2^{(j)} - \alpha_2)\nu_2|$$

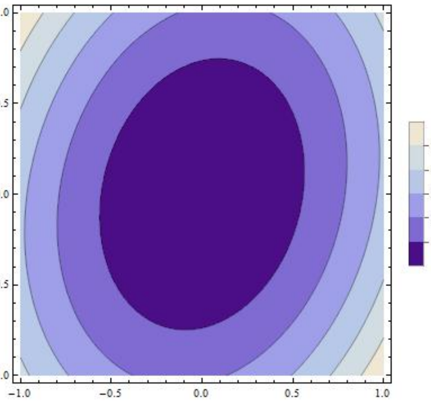

Figure 1: Contour plot for $Q(\alpha)$

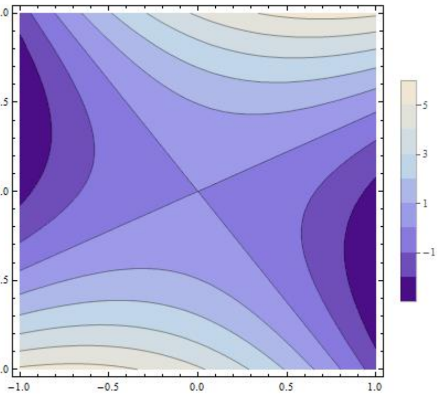

Figure 2: Contour plot for $Q(\beta)$

where $\nu_i \in [-1,1]$ for $i = 1,2$. We see that the maximum occurs in one of the corners where $(\nu_1,\nu_2) = \pm(1,-1)$ and this maximum takes the value 5.84287. So $c(\boldsymbol{\alpha}) \approx \frac{1}{5.84287} \approx 0.17115$.

Example 3. The function $f(x) = x^3 + x^2 - 2x - 1$ has three real roots, namely $2\cos\frac{2\pi}{7}, 2\cos\frac{4\pi}{7}, 2\cos\frac{6\pi}{7}$. Define $F = Q(\beta)$, where $\beta = 2\cos\frac{2\pi}{7}$. Then $(1,\beta,\beta^2)$ is a basis for the number field F of degree 3 with $r = 2$ and $s = 0$. Then $\Sigma = \{\pm1,\pm1\}$. Define $(\beta_1,\beta_2) = (\beta,\beta^2)$, $\beta^{(1)} = 2\cos\frac{4\pi}{7}$ and $\beta^{(2)} = 2\cos\frac{6\pi}{7}$. In figure 2 we see the contour plot of

$$\prod_{j=1}^{2} |(\beta_1^{(j)} - \beta_1)\nu_1 + \ldots + (\beta_2^{(j)} - \beta_2)\nu_2|$$

where $\nu_1,\nu_2 \in [-1,1]$. We see that there are indeed four different segments for each $\sigma \in \Sigma$. Again we calculate the maximum and find $c(\boldsymbol{\alpha}) \approx \frac{1}{5.335605} = 0.18742$.

2.2 The unit conjecture

We shall prove the unit conjecture for real number fields of degree 3. We also tried to prove it for number fields of higher degree, but we did not manage to do this. We will describe the problem we encountered and formulate a new conjecture.

Proof of the unit conjecture for a totally real number field of degree 3. Suppose F is a totally real number field of degree 3, that is $r = 2$ and $s = 0$. By Dirichlets unit theorem, each unit $\eta \in F$ can be written as $\eta = \pm\psi_1^{k_1} \cdot \psi_2^{k_2}$ where ψ_1 and ψ_2 are fundamental units. Now we want to prove that for each $(a_1,a_2) \in \mathbb{R}^2$ with $a_1,a_2 > 0$ and any $\epsilon > 0$ there exists a unit η such that

$$\frac{\eta^{(2)}}{\eta^{(1)}} = \frac{(\psi_1^{k_1} \cdot \psi_2^{k_2})^{(2)}}{(\psi_1^{k_1} \cdot \psi_2^{k_2})^{(1)}} = \frac{a_2}{a_1} + \epsilon.$$

Taking logarithms gives us the equation

$$k_1 \log\frac{\psi_1^{(2)}}{\psi_1^{(1)}} + k_2 \log\frac{\psi_2^{(2)}}{\psi_2^{(1)}} \approx \log\frac{a_2}{a_1}.$$

Since the set $\{m\alpha+n\beta : m,n \in \mathbb{Z}\}$ is dense in $\mathbb{R}$ if and only if α and β are $\mathbb{Z}$-linearly independent, we are left to prove that for $m_1, m_2 \in \mathbb{Z}$ we have

$$m_1 \log \frac{\psi_1^{(2)}}{\psi_1^{(1)}} + m_2 \log \frac{\psi_2^{(2)}}{\psi_2^{(1)}} = 0 \text{ if and only if } m_1 = m_2 = 0.$$

Suppose $m_1 \log \frac{\psi_1^{(2)}}{\psi_1^{(1)}} + m_2 \log \frac{\psi_2^{(2)}}{\psi_2^{(1)}} = 0$ for some $m_1, m_2 \in \mathbb{Z}$, then

$$\left(\frac{\psi_1^{(2)}}{\psi_1^{(1)}}\right)^{m_1} \left(\frac{\psi_2^{(2)}}{\psi_2^{(1)}}\right)^{m_2} = 1.$$

This means that $\eta = \psi_1^{m_1}\psi_2^{m_2}$ is a unit for which $\eta^{(2)} = \eta^{(1)}$. Now let $f^\eta(x)$ be the minimal polynomial of η, then f^η has a double root, hence the roots of f^η are integers. This is only possible when $\eta = \pm 1$. □

Proof of the unit conjecture for a real number field with $r = 0$ and $s = 1$. Let F be a real number field of degree 3 with one real and two complex embeddings. By Dirichlets unit theorem, there exists a fundamental unit ψ such that each unit $\eta \in F$ is of the form $\pm\psi^k$ with $k \in \mathbb{Z}$. Now we want to prove that for all complex numbers a_1, a_2 with $a_1 = \overline{a_2}$ and any $\epsilon > 0$ there exists a unit $\eta = \pm\psi^k \in F$ such that

$$\frac{\left(\psi^{(2)}\right)^k}{\left(\psi^{(1)}\right)^k} = \frac{a_2}{a_1} + \epsilon' \text{ with } |\epsilon'| < \epsilon.$$

Since $\psi^{(1)}$ and $\psi^{(2)}$ are complex conjugates, we have that $\left|\frac{\psi^{(2)}}{\psi^{(1)}}\right| = 1$ hence $\theta = \frac{\psi^{(2)}}{\psi^{(1)}}$ lies on the unit circle. Now θ^k lies on the unit circle for all $k \in \mathbb{Z}$ and since F allows a real embedding there are no roots of unity. That means that $\theta^{k_1} \neq \theta^{k_2}$ when $k_1 \neq k_2$ thus $\{\theta^k : k \in \mathbb{Z}\}$ lies dense on the unit circle. So there is exists a $k \in \mathbb{Z}$ such that $\frac{\left(\psi^{(2)}\right)^k}{\left(\psi^{(1)}\right)^k}$ lies arbitrarily close to $\frac{a_2}{a_1}$. □

The unit conjecture for real number fields of higher degree. Suppose F is a real number field of degree $n+1$, with $r+1$ real and $2s$ complex embeddings. By Dirichlets unit theorem we know that there exist $r+s$ fundamental units $\psi_1, \ldots, \psi_{r+s}$ such that each unit $\eta \in O_F^*$ is of the form $\eta = \psi_1^{k_1}\cdots\psi_{r+s}^{k_{r+s}}$. We want to prove that for each $(a_1, \ldots, a_n) \in \mathbb{C}_+^{r,s}$ and any $\epsilon > 0$ there exists a unit $\eta = \psi_1^{k_1}\cdots\psi_{r+s}^{k_{r+s}}$ such that

$$\frac{(\psi_1^{k_1}\cdots\psi_{r+s}^{k_{r+s}})^{(j)}}{(\psi_1^{k_1}\cdots\psi_{r+s}^{k_{r+s}})^{(1)}} = \frac{a_j}{a_1} + \epsilon_j \text{ with } |\epsilon_j| < \epsilon \text{ for } j = 2, \ldots, n+1.$$

When we take logarithms this translates to

$$k_1 \log \frac{\psi_1^{(j)}}{\psi_1^{(1)}} + \ldots + k_{r+s} \log \frac{\psi_{r+s}^{(j)}}{\psi_{r+s}^{(1)}} \approx \log \frac{a_j}{a_1} \text{ for } j = 2, \ldots, n.$$

These sums are real for $j = 2, \ldots, r$ and complex for $j = r+1, \ldots, s$. Now this can be written as

$$k_1 \log \left|\frac{\psi_1^{(j)}}{\psi_1^{(1)}}\right| + \ldots + k_{r+s} \log \left|\frac{\psi_{r+s}^{(j)}}{\psi_{r+s}^{(1)}}\right| \approx \log \left|\frac{a_j}{a_1}\right| \text{ for } j = r+1, \ldots, r+s.$$

$$k_1 \log \arg(\frac{\psi_1^{(j)}}{\psi_1^{(1)}}) + \ldots + k_{r+s} \log \arg(\frac{\psi_{r+s}^{(j)}}{\psi_{r+s}^{(1)}}) + k_{s+j}2\pi \approx \log \arg(\frac{a_j}{a_1}) \text{ for } j = r+1, \ldots, r+s.$$

Note that $\left|\frac{\psi_1^{(r+j)}}{\psi_1^{(1)}}\right| = \left|\frac{\psi_1^{(j)}}{\psi_1^{(1)}}\right|$ and $\arg(\frac{\psi_1^{(r+j)}}{\psi_1^{(1)}}) = -\arg(\frac{\psi_1^{(j)}}{\psi_1^{(1)}})$ for $j = r+s+1, \dots n+1$ hence we can omit these embeddings in this system of equations. Now we have $r+2s-1$ equations in $r+2s$ integer unknowns. The $(r+2s-1) \times (r+2s)$-matrix corresponding to the left hand sides of this equation is

$$M = \begin{pmatrix} \log \frac{\psi_1^{(2)}}{\psi_1^{(1)}} & \dots & \log \frac{\psi_{r+s}^{(2)}}{\psi_{r+s}^{(1)}} & 0 & 0 & \dots & 0 \\ \vdots & & & \vdots & & & \vdots \\ \log \frac{\psi_1^{(r)}}{\psi_1^{(1)}} & \dots & \log \frac{\psi_{r+s}^{(r)}}{\psi_{r+s}^{(1)}} & 0 & 0 & \dots & 0 \\ \log \left|\frac{\psi_1^{(r+1)}}{\psi_1^{(1)}}\right| & \dots & \log \left|\frac{\psi_{r+s}^{(r+1)}}{\psi_{r+s}^{(1)}}\right| & 0 & \dots & 0 & \\ \vdots & & & \vdots & \vdots & & \\ \log \left|\frac{\psi_1^{(r+s)}}{\psi_1^{(1)}}\right| & \dots & \log \left|\frac{\psi_{r+s}^{(r+s)}}{\psi_{r+s}^{(1)}}\right| & 0 & \dots & 0 & \\ \log \arg(\frac{\psi_1^{(r+1)}}{\psi_1^{(1)}}) & \dots & \log \arg(\frac{\psi_{r+s}^{(r+1)}}{\psi_{r+s}^{(1)}}) & 2\pi & 0 & \dots & 0 \\ \vdots & & & \vdots & & & \vdots \\ \log \arg(\frac{\psi_1^{(r+s)}}{\psi_1^{(1)}}) & \dots & \log \arg(\frac{\psi_{r+s}^{(r+s)}}{\psi_{r+s}^{(1)}}) & 0 & 0 & \dots & 2\pi \end{pmatrix}.$$

Now we are left to prove that the $\mathbb{Z}$-span of the columns of M lies dense in $\mathbb{R}^{r+2s-1}$. We did not manage to this and we conjecture that this is the case. In Proposition 2.8 we formulate the conditions that have to be met. For the proof of this proposition we need Theorem 2.7.

Theorem 2.7 (Kronecker's approximation theorem). *Let $\theta_1, \dots, \theta_n \in \mathbb{R}^n$ be arbitrary real numbers. Suppose that the real numbers $1, \alpha_1, \dots, \alpha_n$ are linearly independent over $\mathbb{Q}$ and that $\epsilon > 0$ is given. Then there exists an integer $k \in \mathbb{Z}$ such that*

$$||k\alpha_i - \theta_i|| < \epsilon \text{ for } i = 1, \dots, n.$$

Proof. This is Theorem 7.10 of [9]. □

Proposition 2.8. *Let P be a $n \times (n+1)$ matrix with columns $\boldsymbol{b}_i \in \mathbb{R}^n$ for $i = 1, \dots, n+1$. Suppose that all $n \times n$ sub-determinants are non-zero and are linearly independent over $\mathbb{Q}$, then the $\mathbb{Z}$-span of the columns of P lies dense in $\mathbb{R}^n$.*

Proof. By the assumptions, it is possible to perform a coordinate transformation on the $\boldsymbol{b}_i$ to write $\boldsymbol{b}_i = \boldsymbol{e}_i$ for $i = 1, \dots, n$ (where $\{\boldsymbol{e}_i, 1 \leq i \leq n\}$ is the standard basis) and $\boldsymbol{b}_{n+1} = (v_1, \dots, v_n)$ where $1, v_1, \dots, v_n$ are linearly independent over $\mathbb{Q}$. Then the propositions follows directly from Theorem 2.7. □

3 Reduction of quadratic forms

In the next section we will describe an algorithm to find integers $(p_1, \ldots, p_n)$ such that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_2^n \tag{7}$$

is small for a given $\boldsymbol{\alpha} \in \mathbb{R}^n$. This algorithm is based on the reduction of quadratic forms, hence in this section we shall introduce the notion of a quadratic form and we will describe two reduction processes, namely Minkowski reduction and LLL-reduction.

3.1 Quadratic forms

A *quadratic form* in n variables is defined as

$$Q(\boldsymbol{x}) = \sum_{i,j=1,\ldots,n} q_{ij}x_ix_j,$$

where $q_{ij} = q_{ji}$ for $1 \leq i < j \leq n$. Such a form is called *positive definite* if $Q(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$ and $Q(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} = 0$. With each quadratic form we associate a matrix $Q = (q_{ij})_{1\leq i,j\leq n}$. The absolute value of the determinant of this matrix is called the determinant of the form, denoted by $D(Q)$. We say that two forms Q and $\tilde{Q}$ are equivalent if there exists a matrix $g \in GL(n, \mathbb{Z})$ such that $Q(g\boldsymbol{x}) = \tilde{Q}(\boldsymbol{x})$. Further, we define $\mu(Q)$ to be the minimum value of $Q(\boldsymbol{x})$, taken over all non-zero $\boldsymbol{x} \in \mathbb{Z}^n$ and $\mu_i(Q)$ to be the smallest value ρ such that there are exactly i independent $\boldsymbol{x} \in \mathbb{Z}^n$ such that $Q(\boldsymbol{x}) \leq \rho$. A theorem of Hermite gives us an upperbound for $\mu(Q)$ in terms of the determinant $D(Q)$. A proof can be found in Cassels [10].

Theorem 3.1. *Suppose $Q(\boldsymbol{x})$ is a positive definite form in n variables and let $\mu(Q)$ be the minimum non-zero value of the set $\{Q(\boldsymbol{x})|\boldsymbol{x} \in \mathbb{Z}^n\}$, there exists a γ_n such that*

$$\mu(Q) \leq \gamma_n D(Q)^{1/n} \text{ where } \gamma_n \leq 2n/3.$$

This minimal value of a quadratic form is of importance since this will help us to find small values for (7). To find the minimal value of a form we need a reduction procedure for quadratic forms.

3.2 Minkowski reduction

Definition 3.2. A quadratic form $Q(\boldsymbol{x}) = \sum_{i,j=1,\ldots,n} q_{ij}x_ix_j$ is called *Minkowski reduced* if for $1 \leq i \leq n$ we have

$$Q(\boldsymbol{e}_i) \leq Q(\boldsymbol{m}) \text{ for all } \boldsymbol{m} \in \mathbb{Z}^n \text{ with } \gcd(m_i, \ldots, m_n) = 1.$$

Another way to describe these inequalities is the following.

$$0 < q_{11} \leq q_{22} \leq \ldots \leq q_{nn} \text{ and } Q(\boldsymbol{y}) \geq q_{mm} \text{ for } 1 \leq m \leq n,$$

where $\boldsymbol{y} \in \mathbb{Z}^n$ is such that $y_i \in \{-1, 0, 1\}$ for $i < m$, $y_m = 1$ and $y_i = 0$ for $i > m$. Hence a form is Minkowski reduced when its coefficients satisfy a finite number of linear inequalities.

Example 4. When $n = 2$ these inequalities are

$$\begin{aligned}0 &\leq q_{11} \leq q_{22}\\ |2q_{12}| &\leq q_{11}.\end{aligned}$$

For $n = 3$ the inequalities are

$$\begin{aligned}&0 \leq q_{11} \leq q_{22} \leq q_{33}\\ &|2q_{12}| \leq q_{11}\\ &|2q_{13} \leq q_{11}\\ &|2q_{23}| \leq q_{22}\\ &q_{11} + q_{22} + 2q_{12} + 2q_{13} + 2q_{23} \geq 0\\ &q_{11} + q_{22} - 2q_{12} - 2q_{13} + 2q_{23} \geq 0\\ &q_{11} + q_{22} - 2q_{12} + 2q_{13} - 2q_{23} \geq 0\\ &q_{11} + q_{22} + 2q_{12} - 2q_{13} - 2q_{23} \geq 0.\end{aligned}$$

When a form Q is Minkowski reduced, we have that $\mu(Q) = Q(\boldsymbol{e_1}) = q_{11}$. The advantage of Minkowski reduction is that there are only finitely many linear inequalities that have to be met, the disadvantage is that the number of inequalities grows exponentially with the dimension, so for higher dimensions this becomes very unpractical.

3.3 LLL-reduction

Another way to reduce a quadratic form is by use of the LLL-algorithm, which is named after its inventors Lenstra, Lenstra and Lovasz [5]. To define this LLL-reduction we first write $Q(\boldsymbol{x})$ in recursive form, that is

$$\begin{aligned}Q(\boldsymbol{x}) =& b_1(x_1 + \mu_{12}x_2 + \ldots + \mu_{1n}x_n)^2\\ &+ b_2(x_2 + \mu_{23}x_3 + \ldots + \mu_{2n}x_n)^2\\ &\vdots\\ &+ b_{n-1}(x_{n-1} + \mu_{n-1n}x_n)^2 + b_n x_n^2.\end{aligned} \tag{8}$$

Definition 3.3. Fix an $\omega \in [3/4, 1]$. A positive definite quadratic form $Q(\boldsymbol{x})$ is called LLL-reduced if

$$\begin{aligned}&1.\quad |\mu_{ij}| \leq 1/2 \text{ for all } i < j.\\ &2.\quad \omega b_i \leq b_{i+1} + \mu_{i,i+1}^2 b_i \text{ for all } i < n. \text{ (Lovasz condition)}\end{aligned} \tag{9}$$

Hermite formulated a notion of reduction for a quadratic form as follows. A quadratic form Q, written in recursive form is reduced if

- $n = 1$.
- When $n > 1$ we have $b_1 = \mu(Q), |\mu_{1j}| \leq 1/2$ for $j = 2, \ldots, n$ and the form $Q - b_1(x_1 + \mu_{12}x_2 + \cdots + \mu_{1n}x_n)^2$ in $x_2, \ldots, x_n$ is reduced.

From this notion of reducedness it follows that $b_i \leq b_{i+1} + \mu_{i,i+1}^2 b_i$ for $i = 1, \ldots, n-1$, thus, when $\omega < 1$, Lovasz condition is a relaxed version of this condition.

The advantage of LLL-reduction in comparison to Minkowski reduction is that the number of conditions that have to be met is relatively small, even in large dimension. The disadvantage is that we do not find optimal results, as the following theorem shows.

Theorem 3.4. *Let Q by an LLL-reduced positive definite form in n variables, with $\omega = 3/4$. Then*

1. $D(Q) \leq \prod_{i=1}^{n} Q(\boldsymbol{e}_i) \leq 2^{n(n-1)} D(Q)$.
2. $Q(\boldsymbol{e}_1) \leq 2^{(n-1)/2} D(Q)^{1/n}$.
3. $Q(\boldsymbol{e_1}) \leq 2^{n-1} \mu(Q)$.
4. *For $k = 1, \ldots, n$ and all $j \leq k$ we have*
$$Q(\boldsymbol{e}_j) \leq 2^{n-1} \mu_k(Q).$$

Proof. First we prove 1. Note that $Q(\boldsymbol{e}_i) = b_i + b_{i-1}\mu_{i-1,i}^2 + \ldots + b_1\mu_{1i}^2$, that $D(Q) = \prod_i^n b_i$ and we can rewrite Lovasz condition as $b_i \geq (\omega - \mu_{i,i+1}^2)b_{i-1}$. Since $\omega = 3/4$ and $|\mu_{i,i+1}| \leq 1/2$, this gives $b_i \geq \frac{1}{2}b_{i-1}$, so $b_j \geq 2^{i-j}b_i$ for $j \geq i$, hence $b_i \leq 2^{j-i}b_j$. Since $Q(\boldsymbol{e}_i) = b_i + b_{i-1}\mu_{i-1,i}^2 + \ldots + b_1\mu_{1i}^2$ we have

$$\begin{aligned} Q(\boldsymbol{e}_i) &\leq b_1 + \frac{1}{4}(b_{i-1} + \ldots + b_1) \\ &\leq b_1 + \frac{1}{4}(2 + \ldots + 2^{i-1})b_i \leq 2^{i-1}b_i. \end{aligned}$$

Since $D(Q) = \prod_i^n b_i$ this gives

$$D(Q) \leq \prod_i^n Q(\boldsymbol{e}_i) \leq \prod_i^n 2^{i-1}b_i \leq 2^{n(n-1)/2} \prod_i^n b_i = 2^{n(n-1)/2}D(Q).$$

Next we proof 2. Since $Q(e_1) = b_1$ we have by the arguments above that $Q(\boldsymbol{e}_i) \leq 2^{i-1}b_i$ for $i = 1, \ldots, n$, hence

$$\begin{aligned} &Q(\boldsymbol{e}_1)^n \leq \prod_{i=1}^{n} 2^{i-1}b_i = 2^{n(n-1)/2}D(Q) \text{ so} \\ &Q(\boldsymbol{e}_1) \leq 2^{(n-1)/2}D(Q)^{1/n}. \end{aligned}$$

Since 3 is a special case of 4 (with $k = 1$), we are left to prove 4. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ be independent non-zero vectors in $\mathbb{Z}^n$ such that $\max_{i=1,\ldots,k} Q(\boldsymbol{x}_i) = \mu_k(Q)$. Choose l minimal such that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ lie in the span of $\boldsymbol{e}_1, \ldots, \boldsymbol{e_l}$. Since the $\boldsymbol{x}_i$ are independent, we have $l \geq k$. Then for at least one of the $\boldsymbol{x}_i$, the l-th coordinate is non-zero. Suppose this is for $\boldsymbol{x}_i$ and suppose this l-th coordinate is ξ. Then $Q(\boldsymbol{x}_i) \geq b_l\xi^2 \geq b_l$. Then for all $j \leq l$ we have

$$Q(\boldsymbol{e}_j) \leq 2^{l-1}b_l \leq 2^{l-1}Q(\boldsymbol{x}_i) \leq 2^{l-1}\mu_k(Q).$$

Since this holds for all $j \leq l$, it certainly holds for all $j \leq q$, which proves the theorem. □

In the LLL-reduction process we need two operations, namely a *shift* and a *swap*. A shift is of the form $x_r \to x_r + ax_s$ where $s > r$ and $a \in \mathbb{Z}$ and is chosen such that after a shift $|\mu_{rs}| \leq \frac{1}{2}$. A swap is of the form $x_r \leftrightarrow x_{r+1}$. Beukers [7] described the following possible implementation of the LLL-algorithm.

1. Perform all necessary shifts to have $|\mu_{i,i+1}| \leq 1/2$. So, for $i = 1$ to $n-1$, perform a shift $x_i \to x_i + ax_{i+1}$, for the logical choice of a.

2. Now find i such that $b_{i+1} + \mu_{i,i+1}^2 b_i < \omega b_i$ and swap x_i and x_{i+1}. Perform the necessary shifts of the form $x_j \to x_j + a_j x_{j+1}$, for $j = i-1, i, i+1$. Repeat this untill no such i exists.

3. Perform a sequence of shifts $x_i \to x_i + a x_j$ to make sure that $|\mu_{ij}| \leq 1/2$ for all $1 \leq i < j \leq n$.

After step 2, the form is partially LLL-reduced. Suppose we start with the form $Q(\boldsymbol{x})$ in recursive form as described in (8). After a shift or a swap we have a new form $\tilde{Q}$, equivalent to Q, with parameters $\tilde{b_i}$ and $\tilde{\mu_{ij}}$. To implement the algorithm we need to know how these new parameters can be calculated. After performing the shift $x_r \to x_r + a x_s$ with $s > r$, we can compute the new parameters with

1. $\tilde{b}_i = b_i$ for all i.
2. $\tilde{\mu}_{is} = \mu_{is} + a\mu_{ir}$ for $i = 1, \ldots, r-1$.
3. $\tilde{\mu}_{rs} = \mu_{rs} + a$.
4. $\tilde{\mu}_{ij} = \mu_{ij}$ for all other i, j.

After the swap $x_r \leftrightarrow x_{r+1}$, the new parameters are

1. $\tilde{b}_r = b_{r+1} + \mu_{r,r+1}^2 b_r$.
2. $\tilde{b}_{r+1} = b_r b_{r+1} / \tilde{b}_r$.
3. $\tilde{b}_i = b_i$ for all $i \neq r, r+1$.
4. $\tilde{\mu}_{ir} = \mu_{i,r+1}$ for $i < r$.
5. $\tilde{\mu}_{i,r+1} = \mu_{ir}$ for $i < r$.
6. $\tilde{\mu}_{r,r+1} = b_r \mu_{r,r+1} / \tilde{b}_r$.
7. $\tilde{\mu}_{rj} = (b_r \mu_{r,r+1} \mu_{rj} + b_{r+1} \mu_{r+1,j}) / \tilde{b}_r$ for $j > r+1$.
8. $\tilde{\mu}_{r+1,j} = \mu_{rj} - \mu_{r,r+1} \mu_{r+1,j}$ for $j > r+1$.
9. $\tilde{\mu}_{ij} = \mu_{ij}$ for all other i, j.

These updates can be verified by simple calculations.

4 Geodesic continued fraction

In this section we will describe an algorithm to find integer $p_1, \ldots, p_n$ such that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_2^n \tag{10}$$

becomes small. Note that we used the Euclidean norm here instead of the supremum norm which we used in the previous sections. Since $||\boldsymbol{p}||_\infty \leq \sqrt{n} \cdot ||\boldsymbol{p}||_2$ we have that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot ||\boldsymbol{p}||_2^n \leq \sqrt{n}^n \cdot ||p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot ||\boldsymbol{p}||_\infty^n.$$

In 1850 Hermite proposed to use quadratic forms to find simultaneous approximations to rational $(\alpha_1, \ldots, \alpha_n)$. His idea was to use the quadratic form

$$Q_t(\boldsymbol{x}, y) = (x_1 - \alpha_1 y)^2 + \cdots + (x_d - \alpha_n y)^2 + ty^2 \tag{11}$$

for any $t > 0$ and find the set of integers $(p_1, \ldots, p_n, q)$ that minimize Q_t. By letting t decrease to zero, this will give several $(p_1, \ldots, p_n, q)$ for which $\max_{i=1,\ldots,n} |q\alpha_i - p_i|$ is small. Since we are interested in the dual case we will use the form

$$Q_t(\boldsymbol{x}) = t(x_0 + x_1\alpha_1 + \ldots + x_n\alpha_n)^2 + x_1^2 + \ldots + x_n^2 \tag{12}$$

where we let t increase to infinity.

Proposition 4.1. *Let $\boldsymbol{x} \in \mathbb{R}^{n+1}$ and suppose that $(q, \boldsymbol{p}) \in \mathbb{Z}^{n+1}$ minimizes the form*

$$Q_t(\boldsymbol{x}) = t(x_0 + x_1\alpha_1 + \ldots + x_n\alpha_n)^2 + x_1^2 + \ldots + x_n^2.$$

Then

$$|q + p_1\alpha_1 + \ldots p_n\alpha_n| \cdot ||\boldsymbol{p}||_2^n < (n+1)^{n/2}.$$

Proof. The form $Q_t = t(x_0 + \alpha_1 x_1 + \ldots + \alpha_n x_n)^2 + x_1^2 + \ldots + x_n^2$ has determinant t. So by Theorem 3.1 there exists a $q \in \mathbb{Z}$ and $\boldsymbol{p} \in \mathbb{Z}^n$ such that

$$Q_t(q, \boldsymbol{p}) = t(q + p_1\alpha_1 + \ldots + p_n\alpha_n)^2 + ||\boldsymbol{p}||_2^2 \leq \gamma_{n+1} t^{1/(n+1)}.$$

Hence $||\boldsymbol{p}||_2^2 \leq \gamma_{n+1} t^{1/(n+1)}$ and $t(q + p_1\alpha_1 + \ldots + p_n\alpha_n)^2 \leq \gamma_{n+1} t^{1/(n+1)}$. Thus

$$(q + p_1\alpha_1 + \ldots + p_n\alpha_n)^{2/n} \leq \gamma_{n+1}^{1/n} t^{-1/(n+1)}.$$

Their product gives

$$(q + p_1\alpha_1 + \ldots + p_n\alpha_n)^{2/n} ||\boldsymbol{p}||_2^2 \leq \gamma_{n+1}^{1+1/n} \leq \left(\frac{2(n+1)}{3}\right)^{1+1/n} < n+1.$$

From this we conclude

$$|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot ||\boldsymbol{p}||_2^n < (n+1)^{n/2}. \qquad \square$$

This is a factor $(n+1)^{n/2}$ away from Dirichlets bound, but again, since we used the Euclidean norm we would expect a factor $n^{n/2}$ hence this comes very close.

4.1 Geodesic algorithm with LLL-reduction

All that we need now is an algorithm that finds integers that minimize the quadratic form Q_t for varying t. In 1994 Lagarias describes a geodesic algorithm for this. In [4] he describes an algorithm based on Minkowski reduction to find simultaneous approximations with fractions. We slightly modify what he described so that we can use it to find small values of linear forms. Also, as mentioned before, since Minkowski reduction becomes unpractical in higher dimensions, we will use LLL-reduction instead. Let $\boldsymbol{\alpha} \in \mathbb{R}^n$ and without loss of generality we can assume that $|\alpha_i| \leq 1/2$ for all $i = 1, \ldots, n$. Otherwise, just take $\alpha_i = [\alpha_i] - \alpha_i$, where $[\alpha_i]$ denotes the nearest integer to α_i. Now the idea is the following. We start with the quadratic form

$$Q_t^{(0)}(\boldsymbol{x}, y) = t(x_0 + \alpha_1 x_1 + \ldots + \alpha_n x_n)^2 + x_1^2 + \ldots + x_n^2.$$

For $t = 1$, this form is LLL-reduced for any $\omega \leq 1$. Define $P^{(0)}$ as the $(n+1) \times (n+1)$ identity matrix. Now enter the following loop:

1. Determine the maximum of the set $\{t | Q_t^{(k)}$ is LLL-reduced $\}$ and call this maximum t_k.
2. Perform an LLL-reduction on $Q_{t_k+\epsilon}^{(k)}$ for infinitesimal $\epsilon > 0$ and let $A_k \in GL(\mathbb{Z}, n)$ be such that $\boldsymbol{x} \to A_k \boldsymbol{x}$ is the corresponding change of variables.
3. Define $Q_t^{(k+1)}(\boldsymbol{x}) = Q_t^{(k)}(A_k \boldsymbol{x})$ and $P^{(k+1)} = P^{(k)} A_k$.

Now set $(q, \boldsymbol{p}) = P^{(k)} \boldsymbol{e}_1$, i.e. $(q, \boldsymbol{p})$ corresponds to the first column of $P^{(k)}$, then this will give a small value for $|q + \alpha_1 p_1 + \ldots \alpha_n p_n| \cdot ||\boldsymbol{p}||_2^n$ as the following proposition shows.

Proposition 4.2. *Let $Q_t^{(k)}(\boldsymbol{x})$ and $P^{(k)}$ be defined as above, and let $(q, \boldsymbol{p}) = (q, p_1, \ldots, p_n)$ be the first column of $P^{(k)}$, then*

$$|q + \alpha_1 p_1 + \ldots \alpha_n p_n| \cdot ||\boldsymbol{p}||_2^n \leq 2^{n(n+1)/4}.$$

Proof. By Theorem 3.4 (2) and the fact that $\det(Q_t^{(k)}) = t$ we have

$$t(q + \alpha_1 p_1 + \ldots \alpha_n p_n)^2 + ||\boldsymbol{p}||_2^2 \leq 2^{n/2} t^{1/(n+1)}$$

This implies

$$||\boldsymbol{p}||_2^2 \leq 2^{n/2} t^{1/(n+1)} \text{ and } t(q + \alpha_1 p_1 + \ldots \alpha_n p_n)^2 \leq 2^{n/2} t^{1/(n+1)}.$$

Now rewrite the second part as

$$(q + \alpha_1 p_1 + \ldots \alpha_n p_n)^{2/n} \leq 2^{1/2} t^{-1/(n+1)}$$

And their product gives

$$(q + \alpha_1 p_1 + \ldots + \alpha_n p_n)^{2/n} ||\boldsymbol{p}||_2^2 \leq 2^{(n+1)/2}.$$

Hence

$$|q + \alpha_1 p_1 + \ldots \alpha_n p_n| \cdot ||\boldsymbol{p}||_2^n \leq 2^{n(n+1)/4}.$$

□

Remark. As we shall see in the next chapter, the actual values this algorithm finds are much smaller than this $2^{n(n+1)/4}$. Also the other columns of $P^{(k)}$ will give good approximations, sometimes these are even better than the approximation of the first column.

Now we have an algorithm to find integers $\boldsymbol{p} \in \mathbb{Z}^n$ such that

$$||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_2^n$$

is small. The only problem that arises is that the LLL-conditions of Definition 3.3 are quadratic in μ_{ij} and hence polynomial in t. Beukers [7] observed that we can describe these conditions in terms of the determinants of the sub matrices of the matrix $Q_t = (q_{ij})$ corresponding to the quadratic form $Q_t(\boldsymbol{x})$. By doing this, the LLL-conditions become linear in t.

Theorem 4.3. *Let $Q(\boldsymbol{x})$ be a form in n variables and (q_{ij}) the corresponding matrix. Define*

$$B_{ij} = \begin{vmatrix} q_{11} & \cdots & q_{1,i-1} & q_{1j} \\ q_{21} & \cdots & q_{2,i-1} & q_{2j} \\ \vdots & \ddots & \vdots & \vdots \\ q_{i1} & \cdots & q_{i,i-1} & q_{ij} \end{vmatrix}.$$

Then $b_i = B_{i,i}/B_{i-1,i-1}$ for $i = 1, \ldots, n$ (where $B_{00} = 1$) and $\mu_{ij} = B_{ij}/B_{ii}$ for all i, j, with $1 \leq i < j \leq n$.

Proof. We prove this by induction on i. Suppose $Q(\boldsymbol{x}) = \sum_{ij} q_{ij}x_ix_j$ where $q_{ij} = q_{ji}$, then the first part of the recursive form of $Q(\boldsymbol{x})$ looks like

$$q_{11}(x_1 + \frac{q_{12}}{q_{11}}x_2 + \ldots + \frac{q_{1n}}{q_{11}}x_n)^2 + \ldots.$$

Now for $i = 1$ we have by definition $B_{1j} = q_{1j}$ for all $j = 1, \ldots, n$. Then $b_1 = q_{11} = \frac{B_{11}}{B_{00}}$ and $\mu_{1j} = \frac{B_{1j}}{B_{11}} = \frac{q_{1j}}{q_{11}}$. Now suppose that it holds for b_{i-1} and $\mu_{i-1,j}$. We write

$$Q(\boldsymbol{x}) = b_1(x_1 + \mu_{12}x_2 + \ldots + \mu_{1n}x_n)^2 + \tilde{Q}(\boldsymbol{x})$$

where we denote the coefficients of $\tilde{Q}$ by $\tilde{q}_{ij}$ for $2 \leq i \leq j \leq n$. Now $q_{ij} = b_1\mu_{1i}\mu_{1j} + \tilde{q}_{ij}$ where $\mu_{1j} = q_{1j}/q_{11}$ for $j = 2, \ldots, n$ as we have seen above. So

$$\tilde{q}_{ij} = q_{ij} - q_{1j}q_{1i}/q_{11}.$$

We define

$$\tilde{B}_{ij} = \begin{vmatrix} \tilde{q}_{22} & \cdots & \tilde{q}_{2,i-1} & \tilde{q}_{12} \\ \tilde{q}_{22} & \cdots & \tilde{q}_{2,i-1} & \tilde{q}_{2j} \\ \vdots & \ddots & \vdots & \vdots \\ \tilde{q}_{i2} & \cdots & \tilde{q}_{i,i-1} & \tilde{q}_{ij} \end{vmatrix}.$$

Since these are $(i-1) \times (i-1)$ matrices we know by the induction hypothesis that $b_i = \tilde{B}_{ii}/\tilde{B}_{i-1,i-1}$ and $\mu_{ij} = \tilde{B}_{ij}/\tilde{B}_{ii}$. Denote by R_k the k-th row of B_{ij}, then we perform the Gaussian row elimination $R_k \to R_k - \frac{q_{1k}}{q_{11}}R_1$ for $k = 2, \ldots, i$. Then B_{ij} reads

$$B_{ij} \equiv \begin{vmatrix} q_{11} & q_{12} & \cdots & q_{1,i-1} & q_{1j} \\ 0 & \tilde{q}_{22} & \cdots & \tilde{q}_{2,i-1} & \tilde{q}_{2j} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \tilde{q}_{i2} & \cdots & \tilde{q}_{i,i-1} & \tilde{q}_{ij} \end{vmatrix}.$$

Hence $B_{ij} = q_{11}\tilde{B}_{ij}$, which proves the desired formulas for b_i and μ_{ij}. □

Next define C_i to be the sub determinant of $B_{i+1,i+1}$ obtained by deleting the i-th row and i-th column. Then we can reformulate the LLL-conditions of Definition 3.3 as follows.

Proposition 4.4. *Let B_{ij} and C_i be the determinants as defined above. Then the LLL-conditions can be written as*

1. $2|B_{ij}| \leq B_{ii}$ *for all* $1 \leq i < j \leq n$.
2. $\omega B_{i,i} \leq C_i$ *for* $i = 1, \ldots, n-1$.

Proof. That the condition $|\mu_{ij}| \leq \frac{1}{2}$ can be written as $2|B_{ij}| \leq B_{ii}$ immediately follows form the fact that $\mu_{ij} = B_{ij}/B_{ii}$. To prove that the Lovasz-condition can be written as $\omega B_{i,i} \leq C_i$, we need the the Desnanot-Jacobi identity, which is formulated in the following lemma.

Lemma 4.5 (Desnanot-Jacobi). *Suppose M is an $n \times n$ matrix and $\tilde{M}$ is the $(n-2) \times (n-2)$ matrix obtained from M by deletion of the i-th and j-th row and column. By M_{kl} we denote the $(n-1) \times (n-1)$ matrix obtained from M by deletion of the k-th row and l-th column. Then*

$$\det(\tilde{M})\det(M) = \det(M_{ii})\det(M_{jj}) - \det(M_{ij})\det(M_{ji}).$$

Now observe that B_{ii} is the sub determinant of $B_{i+1,i+1}$ obtained by deletion of the $i+1$-th row and i-th column. $B_{i-1,i-1}$ is the sub determinant of $B_{i+1,i+1}$ obtained by deletion of the i-th and $i+1$-th row and column and $B_{i,i+1}$ is the sub determinant of $B_{i+1,i+1}$ by deletion of the $i+1$-th row and i-th column. Also observe that when we transpose the matrix belonging to $B_{i,i+1}$ and delete the i-th row and $i+1$-th column, the determinant is equal to $B_{i,i+1}$ again. Now the Desnanot-Jacobi identity gives us

$$B_{i-1,i-1}B_{ii} = C_i B_{ii} - B_{i,i+1}^2.$$

We can rewrite this to

$$\frac{C_i}{B_{i-1,i-1}} = \frac{B_{i+1,i+1}}{B_{i,i}} + \left(\frac{B_{i,i+1}}{B_{i,i}}\right)^2 \cdot \frac{B_{i,i}}{B_{i-1,i-1}} = b_{i+1} + \mu_{i,i+1}^2 b_i.$$

Also $\omega b_i = \omega \frac{B_{ii}}{B_{i-1,i-1}}$ hence

$$\omega b_i \leq b_{i+1} + \mu_{i,i+1}^2 b_i \text{ is equal to } \omega B_{ii} \leq C_i.$$

□

Now we can rewrite the update rules in terms of the determinants. Let notation be as above and we write $\tilde{B}_{ij}$ and $\tilde{C}_i$ for the sub determinants of $\tilde{Q}$. Suppose we perform the shift $x \to x_r + ax_s$ with $s > r$, then

1. $\tilde{B}_{is} = B_{is} + aB_{ir}$ for $i \leq r$.
2. $\tilde{B}_{ij} = B_{ij}$ for all other i, j.
3. $\tilde{C}_r = C_r + 2aB_{rs} + a^2 B_{rr}$ if $r = s-1$.
4. $\tilde{C}_i = C_i$ whenever $r \neq s-1$ or $r = s-1$ and $i \neq r$.

If we perform the swap $x_r \leftrightarrow x_{r+1}$, then

1. $\tilde{B}_{rr} = C_r$.
2. $\tilde{B}_{ir} = B_{i,i+1}$ for all $i < r$.
3. $\tilde{B}_{i,r+1} = B_{ir}$ for all $i < r$.
4. $\tilde{B}_{r,j} = (B_{r,r+1}B_{r,j} + B_{r-1,r-1}B_{r+1,j})/B_{rr}$ for all $j > r+1$.
5. $\tilde{B}_{r+1,j} = (B_{r+1,r+1}B_{r,j} - B_{r,r+1}B_{r+1,j})/B_{rr}$ for all $j > r+1$.
6. $\tilde{B}_{ij} = B_{ij}$ for all other i, j.
7. $\tilde{C}_r = B_{rr}$
8. $\tilde{C}_{r-1} = (B_{r-2,r-2}C_r + B^2_{r-1,r+1})/B_{r-1,r-1}$ if $r > 1$.
9. $\tilde{C}_{r+1} = (B_{r+2,r+2}C_r + \tilde{B}^2_{r+1,r+2})/B_{r+1,r+1}$ if $r < n-1$.
10. $\tilde{C}_i = C_i$ for all $i \neq r-1, r, r+1$

Proposition 4.6. *Let $\boldsymbol{\alpha} \in \mathbb{R}^n$ and let*

$$Q_t(\boldsymbol{x}) = t(x_0 + \alpha_1 x_1 + \ldots + \alpha_n x_n)^2 + x_1^2 + \ldots + x_n^2.$$

Each of the determinants B_{ij} and C_i as defined above are of the form $ut + v$ where $v \in \mathbb{Z}$, and where u is quadratic in α_i.

Proof. The matrix corresponding to Q_t reads

$$Q = \begin{pmatrix} t & \alpha_1 t & \ldots & \alpha_{n-1}t & \alpha_n t \\ \alpha_1 t & 1+\alpha_1^2 t & \ldots & \alpha_1\alpha_{n-1}t & \alpha_1\alpha_n t \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n-1}t & \alpha_1\alpha_{n-1}t & \ldots & 1+\alpha_{n-1}^2 t & \alpha_{n-1}\alpha_n t \\ \alpha_n t & \alpha_1\alpha_n t & \ldots & \alpha_{n-1}\alpha_n t & 1+\alpha_n^2 t \end{pmatrix}.$$

All 1×1 sub matrices are of the desired form. Write R_j for the j-th row of this matrix and we perform the Gaussian row elimination $R_j \to R_j - \alpha_j R_1$ for $j = 2, \ldots, n$. Then we are left with an upper triangle matrix where t only occurs at position 1,1 and all other elements on the diagonal are 1. Hence this determinant, and all sub determinants are either 1, t, or 0. After a change of variables, for each B_{ij} and C_i we have still have that the k-th row is equal to α_k times the first row, except for the integer part of each entry. By row addition we can manage to write the matrix in such a way that only the first row contains entries with a t (which is a linear combination of the other entries of Q, hence at most quadratic in α_i) and all other rows have only integer entries. Thus these determinants are of the desired form. □

As a consequence of this in combination with the LLL-conditions in Proposition 4.4 we can conclude that the values of $t \geq 1$ for which $Q_t(\boldsymbol{x})$ is LLL-reduced are closed intervals $[t_k, t_{k+1}] \in \mathbb{R}_{\geq 1}$.

Next we look at the update rules, even though the rules for updating the determinants are non-linear, the only non-linear part consists of division by an integer. For example, from update rule 4 after a swap, we deduce

$$\tilde{B}_{rj}B_{rr} = B_{r,r+1}B_{rj} + B_{r-1,r-1}B_{r+1,j}.$$

Now write $B_{rr} = u_0 t + v_0$, $B_{r,r+1} = u_1 t + v_1$, $B_{rj} = u_2 t + v_2$, $B_{r-1,r-1} = u_3 t + v_3$, $B_{r+1,j} = u_4 t + v_4$ and $\tilde{B}_{rj} = u_5 t + v_5$ and expand the left and right hand side to get

$$u_5 u_0 t^2 + (u_5 v_0 + v_5 u_0)t + v_5 v_0 = (u_1 u_2 + u_3 u_4)t^2 + (v_1 u_2 + u_1 v_2 + u_3 v_4 + v_3 u_4)t + v_1 v_2 + v_3 v_4.$$

By comparing the coefficient of t we find

$$u_5 = \frac{v_1 u_2 + u_1 v_2 + u_3 v_4 + v_3 u_4 - v_5 u_0}{v_0}$$

and comparing the constant part gives us

$$v_5 = \frac{v_1 v_2 + v_3 v_4}{v_0}.$$

Since $v_i \in \mathbb{Z}$ by Proposition 4.6, this is indeed only a division by an integer. In a similar way we can show that the only non-linear part of the update rules 5, 8 and 9 consist of division by an integer.

The following theorem shows that the algorithm can detect infinitely many $(p_1, \ldots, p_n)$ for which

$$||p_1 \alpha_1 + \ldots + p_n \alpha_n|| \cdot ||\boldsymbol{p}||_2^n$$

is small if and only if $\{1, \alpha_1, \ldots, \alpha_n\}$ is linearly independent over $\mathbb{Z}$.

Theorem 4.7. *If $\{1, \alpha_1, \ldots, \alpha_n\}$ is linearly independent over $\mathbb{Z}$, then the sequence of critical points $t_0 < t_1 < t_2, \ldots$ is an infinite sequence increasing to ∞. If $\{1, \alpha_1, \ldots, \alpha_n\}$ is linearly dependent over $\mathbb{Z}$, then the sequence terminates.*

For the proof we need the following lemma.

Lemma 4.8. *Let $t_0 > 1$, then the number of $M \in GL(n+1, \mathbb{Z})$ such that $Q_t(M\boldsymbol{x})$ is LLL-reduced for some $1 \leq t < t_0$ is finite.*

Proof. First note that $\mu_i(Q_{t_2}) \leq \mu_i(Q_{t_1})$ if $t_2 \geq t_1 \geq 1$. Now let $t \in [1, t_0)$, then we have by Theorem 3.4 that for each $i = 1, \ldots, n+1$,

$$Q_{t_0}(M\boldsymbol{e}_i) \leq Q_t(M\boldsymbol{e}_i) \leq 2^n \mu_i(Q_t) \leq 2^n \mu_i(Q_1)$$

and there are only finitely many $\boldsymbol{x} \in \mathbb{Z}^{n+1}$ with $Q_t(\boldsymbol{x}) \leq 2^n \mu_i(Q_1)$, hence for each column of M there are only finitely many possibilities. □

Proof of theorem 4.7. Suppose there is a point of accumulation, so there is a t_∞ such that $t_k < t_\infty$ for all k. Then there are infinitely many matrices $P^{(k)}$ such that $Q_{t_k}(P^{(k)}\boldsymbol{x})$ is LLL-reduced. Since the $P^{(k)}$ are distinct for each k, this contradicts Lemma 4.8. Hence either the sequence terminates for some t_k, or it increases to infinity. Suppose it terminates at t_k, then $Q_t(P^{(k)}\boldsymbol{e}_1)$ is LLL-reduced for all $t > t_k$. This is only possible if the coefficient of t is either negative of zero. This coefficient is of the form $(x_0 + x_1\alpha_1 + \ldots + x_n\alpha_n)^2$ hence must be zero. Now define $(q, p_1, \ldots, p_n) = P^{(k)}\boldsymbol{e}_1$, then $q + p_1\alpha_1 + \ldots + p_n\alpha_n = 0$, thus $\{1, \alpha_1, \ldots, \alpha_n\}$ is linearly dependent over the integers. □

Suppose there exists a really good approximation, i.e. for a given $\boldsymbol{\alpha} \in \mathbb{R}^n$, there exists an $n+1$-tuple $(y, \boldsymbol{x}) \in \mathbb{Z}^{n+1}$ such that

$$|y + x_1\alpha_1 + \ldots + x_n\alpha_n| \cdot ||\boldsymbol{x}||_2^n < \epsilon$$

for a very small $\epsilon > 0$. Now let

$$t = \frac{||\boldsymbol{x}||_2^2}{n(y + x_1\alpha_1 + \ldots + x_n\alpha_n)^2}$$

and let P be the matrix that corresponds to this value of t. Then by theorem 3.4 we know that

$$Q_t(P\boldsymbol{e}_1) \leq 2^n(t(y + x\alpha_1 + \ldots + x_n\alpha_n)^2 + ||\boldsymbol{x}||_2^2).$$

Now let $(q, p_1, \ldots, p_n) = P\boldsymbol{e}_1$, then

$$|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \leq t^{-\frac{1}{2}} 2^{\frac{n}{2}} (t(y + x_1\alpha_1 + \ldots + x_n\alpha_n)^2 + ||\boldsymbol{x}||^2)^{\frac{1}{2}} \text{ and}$$
$$||\boldsymbol{p}||_2^n \leq 2^{\frac{n^2}{2}} (t(y + x_1\alpha_1 + \ldots + x_n\alpha_n)^2 + ||\boldsymbol{x}||_2^2)^{\frac{n}{2}}.$$

Hence

$$|q + p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot ||\boldsymbol{p}||_2^n \leq 2^{\frac{n(n+1)}{2}} t^{-\frac{1}{2}} (t(y + x_1\alpha_1 + \ldots + x_n\alpha_n)^2 + ||\boldsymbol{x}||_2^2)^{\frac{n+1}{2}}.$$

For our choice of t this gives,

$$\begin{aligned} |q + p_1\alpha_1 + \ldots + p_n\alpha_n| \cdot ||\boldsymbol{p}||_2^n &\leq \\ 2^{\frac{n(n+1)}{2}} \frac{n^{\frac{1}{2}} |y + x_1\alpha_1 + \ldots + x_n\alpha_n|}{||\boldsymbol{x}||_2} \cdot (\frac{||\boldsymbol{x}||_2^2}{n} + ||\boldsymbol{x}||_2^2)^{\frac{n+1}{2}} &= \\ 2^{\frac{n(n+1)}{2}} n^{\frac{1}{2}} (\frac{1}{n} + 1)^{\frac{n+1}{2}} \cdot |y + x_1\alpha_1 + \ldots + x_n\alpha_n| \cdot ||\boldsymbol{x}||_2^n &\leq \gamma_n\epsilon \end{aligned}$$

where $\gamma_n = 2^{\frac{n(n+1)}{2}} n^{\frac{1}{2}} (\frac{1}{n} + 1)^{\frac{n+1}{2}}$, hence this differs from ϵ by a factor depending only on n.

Remark. Note that the LLL-algorithm on itself is able to detect small values for linear forms. We can start with the quadratic form

$$Q(\boldsymbol{x}) = N(x_0 + \alpha_1 x_1 + \ldots + \alpha_n x_n)^2 + x_0^2 + \ldots + x_n^2$$

for a large integer N and perform an LLL-reduction on this form. Now suppose that $Q(P\boldsymbol{x})$ is reduced for a given $P \in GL(\mathbb{Z}, n+1)$ and let $(q, p_1, \ldots, p_n)$ be a column of P. Then $|q + p_1\alpha_1 + \ldots + p_n\alpha_n|$ is small. We can do this for varying N, for example for $N = 10^k$ where $k = 1, \ldots, 100$. This is much faster than the geodesic algorithm described above, but we might miss the best approximations. With the geodesic algorithm we can divide the whole of $\mathbb{R}^+$ in intervals where each interval corresponds to one reduced form, hence to one transformation matrix $P^{(k)}$ which gives $n + 1$ potentially small linear forms. By trying several values for N we only find a subset of these.

5 Implementation and results

I implemented the algorithm in `Mathematica 9.0`. For this, I used the java code that Harry Smit wrote for the simultaneous approximation of a given $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ with fractions. I chose for Mathematica because the numerical precision of the calculations is much higher in Mathematica than it is in Java. I will give a rough description of how the algorithm is implemented. The code is enclosed as an appendix. All experiments are done on an ASUS laptop with an INTEL CORE i5 processor with 1.80GHz.

5.1 Short description of the code

The module `Start[v]` receives the vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ and calls for `Initialize[v]`, in which the initial values of the determinants B_{ij} an C_i are calculated for $1 \leq i \leq j \leq n+1$. Then `InitialReduction[]` performs the necessary shifts such that the $|\alpha_i| \leq 1/2$ which implies that all inequalities $2|B_{1i}| \leq B_{11}$ are met and thus that the form

$$Q_t(\boldsymbol{x}) = t(x_0 + \alpha_1 x_1 + \ldots + \alpha_n x_n)^2 + x_1^2 + \ldots + x_n^2$$

with $t = 1$ is LLL-reduced for $\omega = \frac{3}{4}$. Now the approximation process can start. We repeatedly call for the module `ApproximationStep[]`, which performs the following steps.

1. `ComputeInterval[]` computes for each inequality $2|B_{ij}| \leq B_{ii}$ for $1 \leq i < j \leq n$ and for each $\omega B_{ii} \leq C_i$ for $i = 1, \ldots, n-1$ the corresponding intervals for t for which the inequality is met.
2. `MakePlan[]` determines the least upper bound of these intervals. In the k-th approximation step, this value corresponds to t_k, which is the maximum of the set $\{t | Q_t^{(k)}$ is LLL-reduced$\}$.
3. If t_k is the upper bound of the interval corresponding to the inequality $2|B_{ij}| \leq B_{ii}$, we call for `Shift[i,j,a]`, which performs the shift $x_i \to x_i + ax_j$ for $a = \pm 1$. If t_k is the upper bound of the interval corresponding to the inequality $\omega B_{ii} \leq C_i$, we call for `Swap[i]`, which performs the swap $x_i \leftrightarrow x_{i+1}$. This value of t_k might not be unique, for a note on that see the remark below.
4. At last we call for `ReductionStep2[]` and `ReductionStep3[]` which perform the necessary shifts and swaps to make the form Q_t reduced for $t = t_k$. These steps correspond to step 2 and 3 of the loop described in Section 4.1.

The combination of those four steps we define to be one *approximation step.* The value t_k we find in step 2 is called a *critical value* for t and the corresponding shift or swap we perform in step 3 we call a *critical shift* or a *critical swap.* We define the shifts and swaps that are performed in step 4 to be *shifts and swaps for reduction.* During the process we keep track of all the changes of variables in the matrix `transform`. This matrix corresponds to $P^{(k)}$ as defined in Section 4.1. Now denote the j-th column of this matrix by $(q_j, p_{1j}, \ldots, p_{nj})$. After each approximation step, the module `ComputeQualityL2[]` computes for $j = 1, \ldots, n+1$ the value

$$|q_j + p_{1j}\alpha_1 + \ldots + p_{nj}\alpha_n| \cdot ||\boldsymbol{p}_{ij}||_2^n.$$

We call this value the *L2-quality* of the approximation. The smaller this value is, the better is the approximation. The module `ComputeQualitySup[]` computes for $j = 1, \ldots, n+1$ the value

$$|q_j + p_{1j}\alpha_1 + \ldots + p_{nj}\alpha_n| \cdot ||\boldsymbol{p}_{ij}||_\infty^n$$

which we call the *Sup-quality* of the approximation. The module `ComputePrecision[]` computes for $j = 1, \ldots, n+1$ the value

$$|q_j + p_{1j}\alpha_1 + \ldots + p_{nj}\alpha_n|$$

which we call the *precision* of the approximation. When $(1, \alpha_1, \ldots, \alpha_n)$ is an integral basis of a number field, the module `ComputeNorm[]` computes the norm of each element $q_j + p_{1j}\alpha_1 + \ldots + p_{nj}\alpha_n$. After each approximation step we store all these values in `table`, together with some counters that keep track of the number of shifts and swaps we have performed up to the given approximation step. We can repeat `ApproximationStep[]` as many times we want, but when the accuracy of t drops below zero the process stops. If we continued the process with a negative accuracy of t, rounding errors take over and the results would no longer be reliable. By increasing the numerical precision (stored in the variable `np`) we can increase the possible number of approximation step.

Remark. Note that it is possible that in step 2 of an approximation step more than one inequality is violated for the same critical value of t_k. If this happens, it is not clear which critical shift or critical swap has to be performed. We did not take in account the possible problems this can cause. The algorithm is programmed in such a way that it automatically chooses the change of variables that corresponds to the first violated inequality it encounters.

5.2 Tests with random α and varying dimension

We tested the algorithm for random vectors $(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ where we varied n from 2 to 25. For this we made a `list` with elements of the form $\sqrt{p_i}$ and $\log p_i$ where p_i runs over all primes from 2 to 541. Then we ran the following.

```
For[dim = 2, dim <= 25, dim++,
 v = RandomSample[list, dim];
 Print[Timing[Start[v]]];
 name = StringJoin["VarDim", StringJoin[ToString[dim], ".xlsx"]]
   Export[name, table]]
```

For each vector, we let the approximation process stop after 1000 approximation steps. Then we exported `table` to excel, hence one run of the previous code produces 24 excel sheets each consisting of 1000 rows with data. We repeated this 5 times, so for each dimension we run the approximation process for 5 different random vectors. With this data we made graphs to get some insight in how well the algorithm performs.

The quality and the precision of the approximation At the end of each approximation step, each of the $n+1$ columns of the transformation matrix $P^{(k)}$ correspond to one linear form in $(\alpha_1, \ldots, \alpha_n)$. Hence in 1000 approximation steps we calculate the quality of $(n+1) \times 1000$ linear forms. Figure 1 shows for each input vector the minimum of all these L2-qualities. Figure 2 shows after how many approximation steps this minimal quality was encountered. Figure 3 and 4 show these results for the sup-quality. Note that we used a logarithmic scale for displaying the qualities. For each n we see five dots, where each dot corresponds to one input vector. We see that when the dimension grows, the L2-quality of the approximation increases and the Sup-quality decreases. Also when $n > 20$ the minimal L2-quality occurs in the beginning of the approximation process, while the minimal Sup-quality occurs in general later in the process.

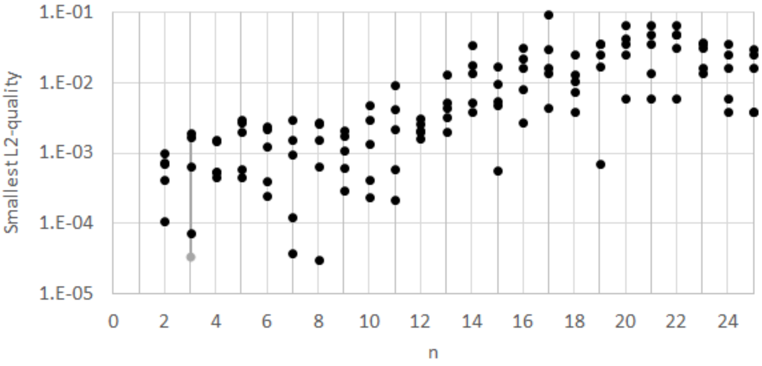

Fig. 1: Minimal L2-quality

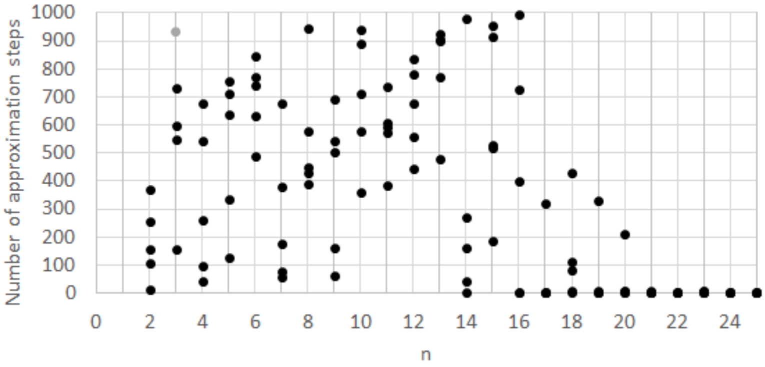

Fig. 2: Number of steps

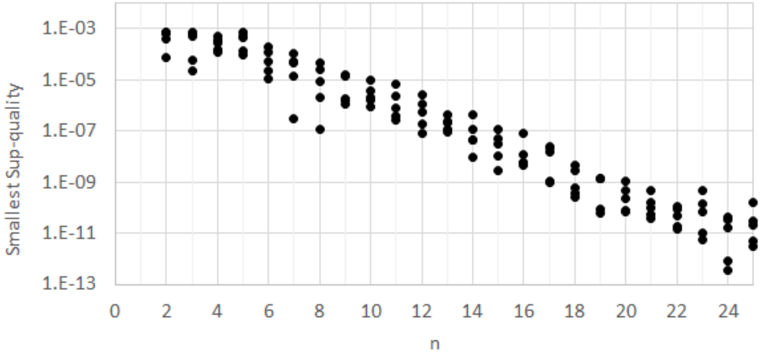

Fig. 3: Minimal Sup-quality

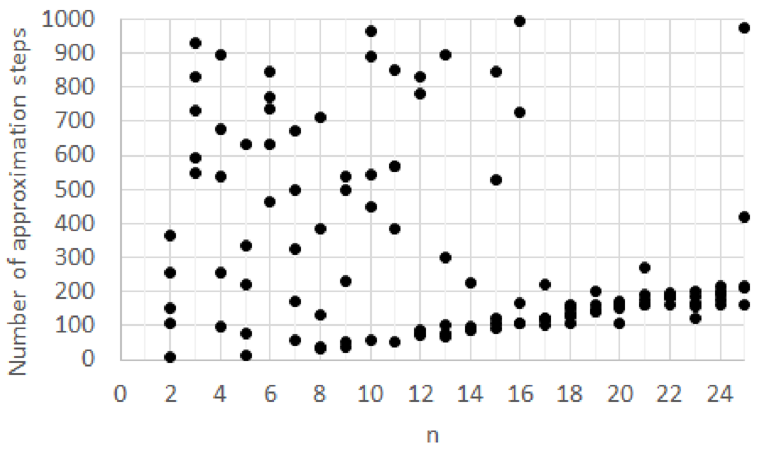

Fig. 4: Number of steps

Example 5. For $n = 3$, one of the input vectors is of the form $(\alpha_1, \alpha_2, \alpha_3) = (\sqrt{257}, \log 89, \log 509)$. After 934 approximation steps the last column of the transformation matrix $P^{(934)}$ looks like

$$\begin{aligned}&(q, p_1, p_2, p_3) =\\ &\quad(-366159353393454310034603422221, -705530129949466352406857980 1,\\ &\qquad 842150824437482219747020229 0, 1795761968888591079355802764 7).\end{aligned}$$

Now $|q + p_1\alpha_1 + p_2\alpha_2 + p_3\alpha_3| \cdot \sqrt{p_1^2 + p_2^2 + p_3^2}^3 = 0.000034038$ which is the smallest L2-quality we encounter in a 1000 approximation steps. This example corresponds to the dark gray dots in Figure 1 and 2.

Example 6. For $n = 20$, one of the input vectors is of the form

$$\begin{aligned}&(\alpha_1, \ldots, \alpha_{20}) =\\ &\quad(\log 277, \sqrt{271}, \sqrt{373}, \sqrt{487}, \log 503, \log 491, \sqrt{449}, \log 97, \sqrt{233}, \sqrt{523},\\ &\qquad \log 131, \log 29, \log 41, \sqrt{239}, \log 107, \log 307, \log 139, \sqrt{433}, \log 461, \sqrt{179})\end{aligned}$$

The smallest L2-quality we encounter is given by the 18th column of the transformation matrix $P^{(1)}$. This column looks like

$$(-5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0)$$

and the corresponding L2-quality is $|-5 + \log(139)| = 0.0655$. Hence this approximation is not of any interest. We do find interesting linear combinations of $(\alpha_1, \ldots, \alpha_{20})$, even though their L2-quality is higher than 0.0655. For example, the first column of $P^{(1000)}$ is

$$\begin{aligned}&(q, p_1, \ldots, p_{20}) = (-5661, 75, 7, 7, 34, -181, 0, 238,\\ &\qquad 51, -19, -110, -78, -11, -57, 29, 100, -19, -53, 88, -16, 93).\end{aligned}$$

Then the L2-quality of this approximation is $|q + p_1\alpha_1 + \ldots + p_{20}\alpha_{20}| \cdot ||\boldsymbol{p}||_2^{20} = 0.27822$. The smallest L2-quality of all columns of $P^{(1000)}$ was given by the first column, but we find the smallest precision in the 18-th column. This column reads

$$\begin{aligned}&(q, p_1, \ldots, p_{20}) = (-14085, 86, 96, 108, -24, -32, 147, -57,\\ &\qquad -171, -8, -29, 21, 127, -103, 322, -78, 120, -216, 413, 132, -94).\end{aligned}$$

Then $|q + p_1\alpha_1 + \ldots + p_{20}\alpha_{20}| = 1.24 \cdot 10^{-54}$.

In the previous example we have seen that the quality of the approximations does not always decrease with the number of approximation steps, but the precision of the approximation does decrease exponentially in the number of approximation steps. We calculated after each approximation step the precision of the $n+1$ linear forms and determined the minimum of these $n+1$ values. For each approximation step, the base 10 logarithm of the minimum of these precisions is displayed in Figure 5. Here we have taken 6 random vectors, namely one for each n where $n = 2, n = 5, n = 10, n = 15, n = 20$ and $n = 25$. All other vectors show similar results. Figure 6 shows the number of digits of the largest element in the transformation matrix $P^{(1000)}$. Again we see see 5 dots for each dimension, each corresponding to one random vector. We see that this number of digits decreases exponentially with the dimension.

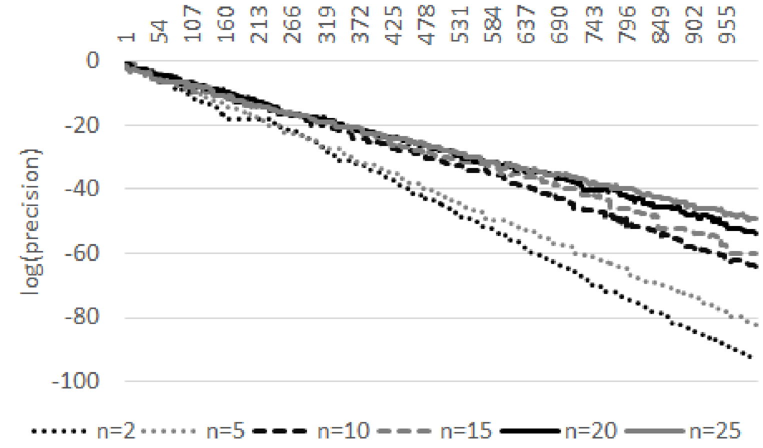

Fig. 5: log(precision)

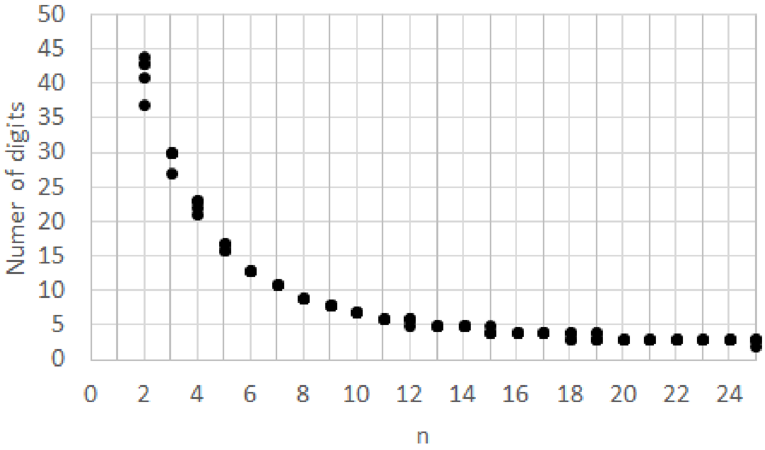

Fig. 6: Number of digits of p_i

Example 7. For $n = 5$ one of the input vectors reads

$$(\alpha_1, \dots, \alpha_5) = (\sqrt{37}, \log 31, \sqrt{19}, \log 61, \sqrt{127}).$$

Figure 5 shows that after 1000 approximation steps the precision of the approximation is approximately 10^{-82} and the largest element of $P^{(1000)}$ has approximately

16 digits. For this specific vector, the 4-th column of $P^{(1000)}$ corresponds to the linear form with the smallest precision. This column reads

$$(q, p_1, \ldots, p_5) = (-549589165745545533, 14913244085348451,$$
$$4168419492155768, 2195766269208450, -4798314965938595, -3541880014481820)$$

and the corresponding precision is $7.25 \cdot 10^{-83}$.

The number of shifts and swaps Apart from the quality and the precision of the approximations, we are also interested in how fast the algorithm finds the approximations. For this we look at the number of shifts and swaps that are performed after each approximation step. To see how these numbers grow with the number of approximation steps, we calculated for $n = 2$, $n = 5$, $n = 15$, and $n = 25$ the average number of shifts and swaps taken over all 5 random vectors. Figure 3, 4, 5 and 6 show how the average number of *critical shifts*, *critical swaps*, *shifts for reduction* and *swap for reduction* grows with the number of approximation steps.

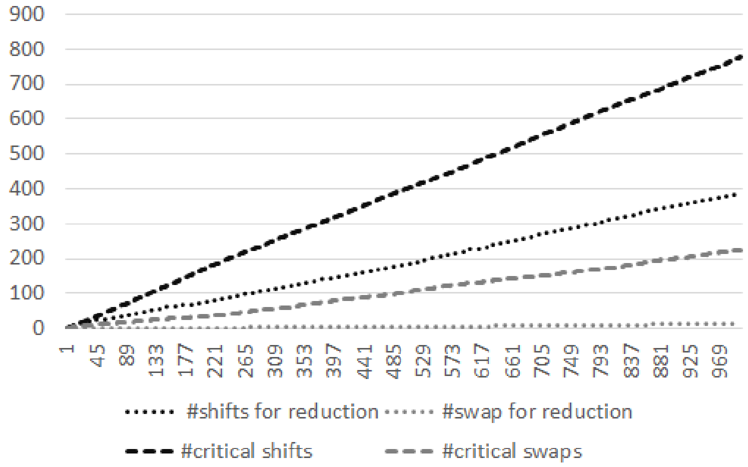

Figure 3: n=2

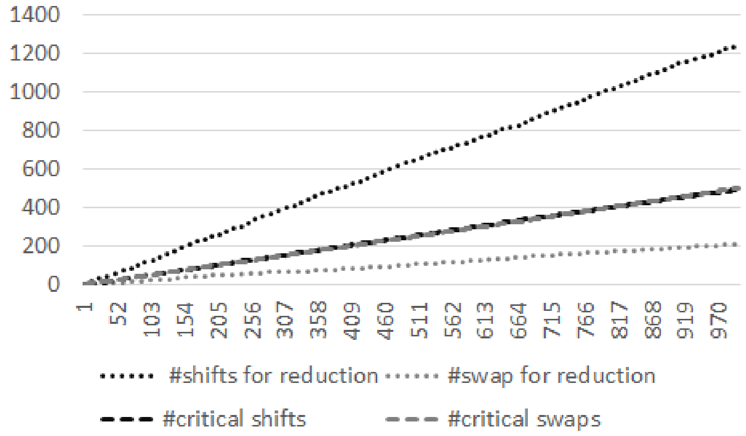

Figure 4: n=5

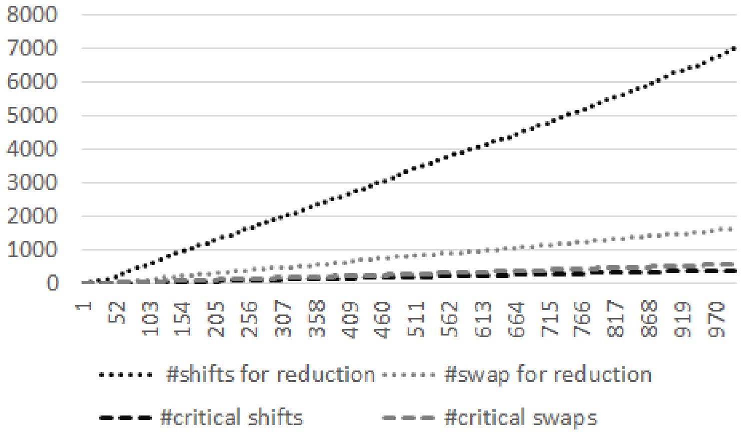

Figure 5: n=15

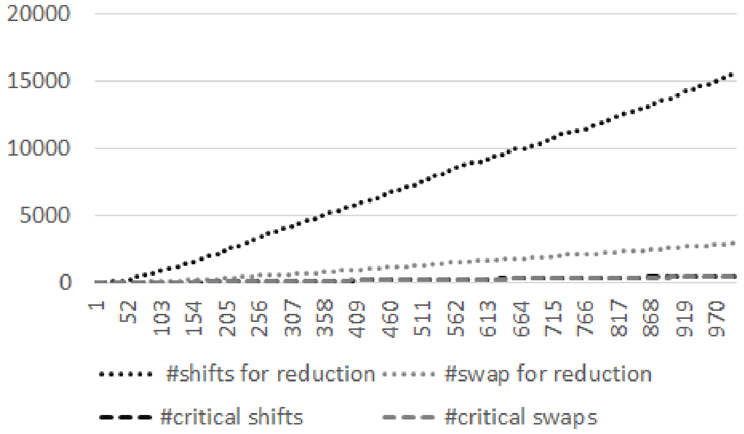

Figure 6: n=25

We see that when $n = 2$ most of the transformations are critical shifts. For $n = 5$ the number of critical shifts is equal to the number of critical swaps, but for $n = 15$ the number of critical shifts drops below the number of critical swaps. Also we see that the number of shifts for reduction grows fast when the dimension grows. To get some more insight in how these values behave for varying n, we looked at the number of shifts and swaps at the end of 1000 approximation steps and displayed these in Figure 7 and 8. Again each dot corresponds to one random input vector.

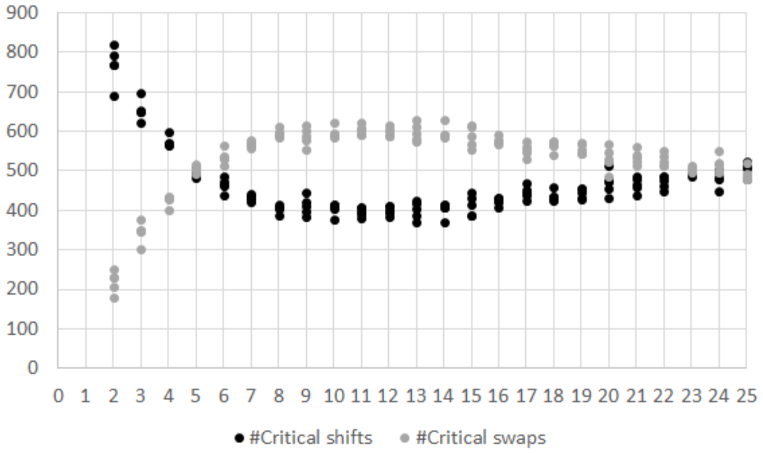

Fig. 7: Critical shifts and swaps

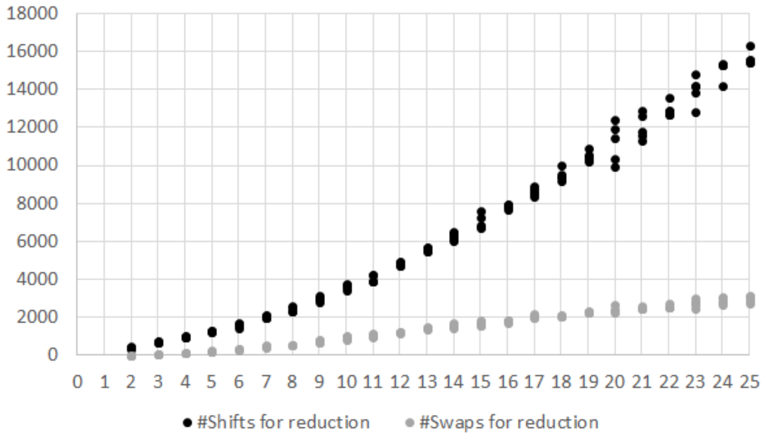

Fig. 8: Shifts and swaps for reduction

We see that for $n \leq 4$ there are more critical shifts than critical swaps. Then, from $n = 6$ to $n = 21$ there are more critical swaps than critical shifts and those numbers stabilize at 500 when n becomes larger. This is unexpected, since there are n inequalities that lead to a critical swap, and $\frac{1}{2}n(n-1)$ inequalities that lead to a critical shift, so we would expect that the number of critical shifts grows with the dimension. In contrary to the number of critical shifts and critical swaps, the number of shifts and swaps for reduction behave as we would expect, namely the number of shifts for reduction grows quadratic in the dimension and the number of swaps for reduction seems to grow linear in the dimension. To decrease the number of shifts for reduction it is possible to work with partial reduction, as defined in Section 3.3. Then only the first column of $P^{(k)}$ will give a good approximation.

Above we have seen that the smallest quality or the smallest precision does not always belong to the first column of the transformation matrix, hence we do not know whether this partial reduction would lead to good results.

Timing For each random input vector we timed the approximation process. Figure 7 shows for how many seconds the algorithm ran for $n = 2$ up to $n = 25$. Again each dot corresponds to one input vector.

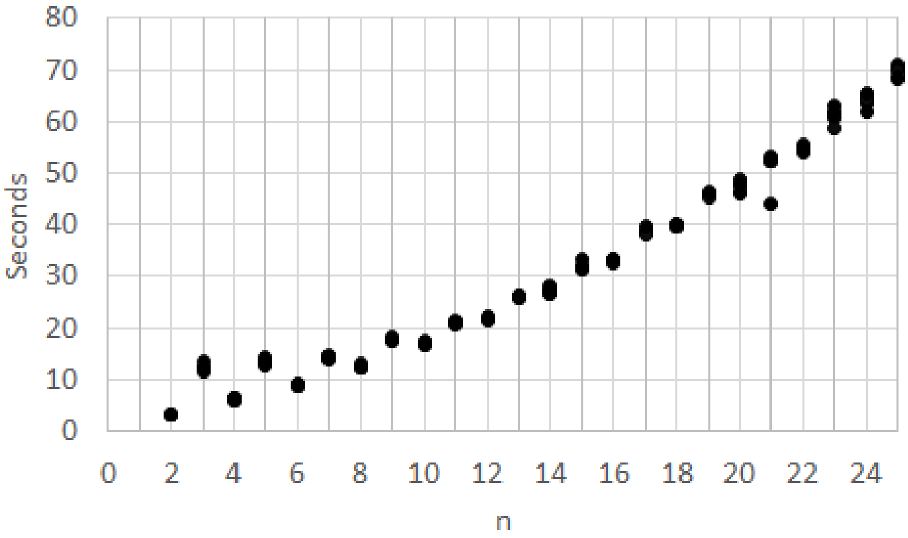

Figure 7: Timing of the approximation process

Here we worked with a numerical precision of 700 and we calculated the qualities and the precisions of the approximations after each approximation step (when we do this at each 10-th step, or only after the last step, the algorithm terminates much faster). Note that for small dimensions, the algorithm terminates faster when the input vector consists of an odd number of elements then when this vector consists of an even number of elements. We do not know why this happens. One can speed up the process by decreasing the numerical precision (but then it might be possible that the algorithm terminates before 1000 steps because the accuracy of t drops below zero).

5.3 Tests with number fields

Let F be a real number field of degree $n+1$ with integral basis $(1, \alpha_1, \ldots, \alpha_n)$. We define

$$c_\infty(\boldsymbol{\alpha}) = \liminf_{||\boldsymbol{p}||_\infty \to \infty} ||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_\infty^n \text{ and}$$
$$c_2(\boldsymbol{\alpha}) = \liminf_{||\boldsymbol{p}||_\infty \to \infty} ||p_1\alpha_1 + \ldots + p_n\alpha_n|| \cdot ||\boldsymbol{p}||_2^n$$

Now we can use Theorem 2.1 to calculate the lower bound for these values. Since all the arguments in the proof of Theorem 2.1 also hold when we replace the supremum norm with the Euclidean norm we can use this theorem to compare the results of our algorithm with the values

$$\min_{\sigma\in\Sigma} \frac{1}{\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_2 = 1 \text{ and } \sigma(A\boldsymbol{\nu}) = \sigma\}} \text{ and}$$
$$\min_{\sigma\in\Sigma} \frac{1}{\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_\infty = 1 \text{ and } \sigma(A\boldsymbol{\nu}) = \sigma\}}.$$

A theorem of Fürtwangler states the following.

Theorem 5.1. *Let D be the smallest discriminant of a real number field of degree $n+1$ and let c be a constant smaller than*

$$\frac{1}{|D|^{\frac{1}{2n}}}.$$

Then there exist $(\alpha_1, \dots, \alpha_n)$ such that there are only finitely many integer solutions to

$$\max_{i=1,\dots,n} |q\alpha_i - p_i| < c \cdot q^{-1/n}.$$

The proof of this theorem can be found in [11]. Because of this theorem we got the idea that an integral basis of a real number field with small discriminant is a logical choice to perform the experiments with. Table 1 shows for number fields of degree $3, 4, 5$ and 6 the smallest discriminant and the corresponding minimal polynomial. For each degree we chose the real number field with smallest discriminant and the totally real number field with smallest discriminant. From now on we will refer to these number fields by stating their discriminant. The number fields with small discriminant and the corresponding minimal polynomials are found in [12], [13], [14], [15] and are listed in Table 1.

$n+1$	$f(\alpha)$	D	$(r+1, s)$
3	$\alpha^3 + \alpha^2 - 1$	-23	$(1,1)$
3	$\alpha^3 + \alpha^2 - 2\alpha - 1$	49	$(3,0)$
4	$\alpha^4 + \alpha^2 - 11$	-275	$(2,1)$
4	$\alpha^4 - 14\alpha^2 + 29$	725	$(4,0)$
5	$\alpha^5 - \alpha^3 + \alpha^2 + \alpha - 1$	1609	$(1,2)$
5	$\alpha^5 + \alpha^4 - 4\alpha^3 - 3\alpha^2 + 3\alpha + 1$	14641	$(5,0)$
6	$\alpha^6 + 2\alpha^5 - 3\alpha^3 + 2\alpha - 1$	28037	$(2,2)$
6	$\alpha^6 + \alpha^5 - 7\alpha^4 - 2\alpha^3 + 7\alpha^2 + 2\alpha - 1$	300125	$(6,0)$

Table 1: Number fields of small discriminant

For each number field we calculated an integral basis in Mathematica with the following command.

```
f = #^3 + #^2 - 1;
v = NumberFieldIntegralBasis[AlgebraicNumber[Root[f, j], {0, 1}]].
```

We did this for each f as denoted in Table 1 and for $j = 1, \dots, r+1$, hence for each possible real embedding. Suppose $(1, \alpha_1, \dots, \alpha_n)$ is a basis for the number field $Q(\alpha)$, then we let the algorithm run for 1000 approximation steps for each input vector $(\alpha_1^{(j)}, \dots, \alpha_n^{(j)})$ for $j = 0, 1, \dots, r$, hence for each real embedding. Again we calculated the minimum of all L2-qualities and the minimum of all Sup-qualities that we encountered in these 1000 steps. These are listed in the fourth and sixth column of Table 2. We used Mathematica to calculate the values $\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_2 = 1\}$ and $\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_\infty = 1\}$. We define

$$\frac{1}{\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_2 = 1\}}$$

to be the *L2CK-constant* and

$$\frac{1}{\max\{\Pi(A\boldsymbol{\nu}) : ||\boldsymbol{\nu}||_\infty = 1\}}$$

to be the *SupCK-constant.* These values are listed in the fifth and seventh column of Table 2. The third column of this table shows the value of α for the several real embeddings.

$n+1$	D	root	Sup-quality	SupCK	L2-quality	L2CK
3	23	0.754878	0.171214	0.171149	0.245122	0.30086
3	49	-1.80194	0.048711	0.047875	0.095571	0.095545
3	49	-0.44504	0.187420	0.187420	0.198062	0.220282
3	49	1.24698	0.187420	0.187420	0.191833	0.191832
4	-275	1.68941	0.006368	0.004181	0.021071	0.019774
4	-275	-1.68941	0.013067	0.009702	0.033157	0.032230
4	725	3.38705	0.000936	0.000527	0.001383	0.001376
4	725	-3.38705	0.001851	0.000838	0.002699	0.002646
4	725	1.58993	0.004464	0.004317	0.009992	0.009540
4	725	-1.58993	0.005552	0.004360	0.011694	0.010905
5	1609	-1.68941	0.012214	0.009644	0.087622	0.085547
5	14641	-1.91899	0.000271	$3.7 \cdot 10^{-6}$	0.000434	$3.0 \cdot 10^{-5}$
5	14641	-1.30972	0.000714	0.000220	0.004243	0.001572
5	14641	-0.28463	0.001379	0.000574	0.004431	0.003162
5	14641	0.83083	0.001783	0.001129	0.006939	0.005203
5	14641	1.68251	0.000171	0.000026	0.001081	0.000208
6	28037	-1.23795	0.000195	0.000042	0.001999	$1.2 \cdot 10^{-6}$
6	28037	0.807788	0.002654	0.000181	0.010987	0.00442
6	300125	-2.9156	$5.1 \cdot 10^{-5}$	$2.8 \cdot 10^{-13}$	0.000306	$1.6 \cdot 10^{-12}$
6	300125	-0.770676	0.000048	0.000029	0.000958	0.000480
6	300125	-0.720093	0.000128	0.000035	0.001553	0.000586
6	300125	0.275051	0.000229	$6.8 \cdot 10^{-6}$	0.001574	0.000110
6	300125	1.11366	0.000158	$2.8 \cdot 10^{-7}$	0.000697	$6.1 \cdot 10^{-6}$
6	300125	2.01766	$4.8 \cdot 10^{-5}$	$2.9 \cdot 10^{-10}$	0.000435	$2.3 \cdot 10^{-9}$

Table 2: The minimum of the qualities and $c(\boldsymbol{\alpha})$

Remark. Note that in the definition of the L2CK-constant and the SupCK-constant we omit the constant N_σ as stated in Theorem 2.1, so it might be possible that these constants lie a factor away from the actual value of $c(\boldsymbol{\alpha})$.

In Figure 8 we displayed for each number field the values $\frac{L2-quality}{L2CK-constant}$. The closer this values lies to 1 the better the approximation is. We see that up to the number field with discriminant 1609 these values lie close to 1. For the number fields with larger discriminant the approximation becomes worse. We only displayed the ratios up to 3, but for the number field with discriminant 300125 this ratio becomes even larger than $1.8 \cdot 10^5$.

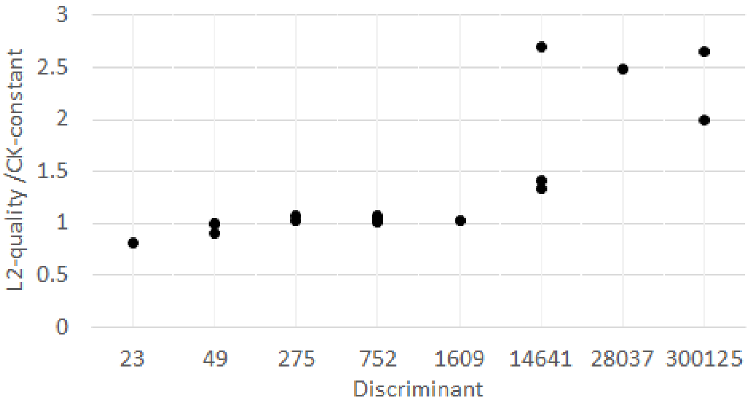

Figure 8: Quality-L2 / L2CK-constant

Elements with small norm Let F be a number field with integral basis $(1, \alpha_1, \ldots, \alpha_n)$, then we can run the algorithm with input vector $(\alpha_1, \ldots, \alpha_n)$ to detect elements with small norm in the number field F. At the end of each approximation step, each column of the transformation matrix $P^{(k)}$ corresponds to an element of the number field F. We define the set

$$\Omega_F = \{q + p_1\alpha_1 + \ldots + p_n\alpha_n : (q, \ldots, p_n) \text{ is a column of } P^{(k)} \text{ for } k = 1, \ldots, 1000\}.$$

Thus Ω_F contains all elements of F that are detected by the algorithm. Next we define for $a \in \mathbb{N}$

$$\Omega_{F,a} = \{\eta : \eta \in \Omega_F \text{ and } |N(\eta)| = a\}.$$

Then, for example, $\Omega_{F,1}$ is the set of all units the algorithm finds. By $|\Omega_F|$ and $|\Omega_{F,a}|$we denote the cardinality of these sets. Note that $|\Omega_F|$ is at most $1000 \times (n+1)$ since at the end of each approximation step each column of the $(n+1) \times (n+1)$ matrix $P^{(k)}$ corresponds to one element of F. We calculated for each number field listed in Table 1 how many distinct elements it detects in 1000 approximation steps and how many of them are units. The results are listed in Table 3. The third column gives the value of α in the chosen real embedding. The last column of this table shows the maximum absolute value of all norms of elements of Ω_F.

$n+1$	D	*root*	$\lvert\Omega_F\rvert$	$\lvert\Omega_{F,1}\rvert$	Maximum norm
3	−23	0.75488	979	979	1
3	49	−1.80194	728	604	13
3	49	−0.44504	778	778	1
3	49	1.24698	807	807	1
4	−275	−1.68941	851	574	29
4	−275	1.68941	841	494	71
4	725	−3.38705	893	326	1331
4	725	−1.58993	859	543	149
4	725	1.58993	860	515	251
4	725	3.38705	866	256	3319
5	1609	−1.68941	853	837	17
5	14641	−1.91899	955	152	167683
5	14641	−1.30972	908	360	989
5	14641	−0.28463	932	498	473
5	14641	0.83083	928	517	571
5	14641	1.68251	932	207	19801
6	28037	−1.23795	999	291	79699
6	28037	0.80779	983	662	211
6	300125	−2.91560	1021	8	$7.98 \cdot 10^{13}$
6	300125	−0.77068	1076	654	6581
6	300125	−0.72009	1045	655	3079
6	300125	0.27505	1067	289	87641
6	300125	1.11366	1046	177	5630339
6	300125	2.01766	1062	69	26795416871

Table 3: Number of units detected by the algorithm

We see that almost all elements of Ω_F are units when F is of degree 3 and the number fields of smaller discriminant lead to more elements with small norm than the number fields of higher discriminants, which was to be expected. A remarkable result is that the choice of embedding is of great influence on the performance of the algorithm. For example, look at $F = Q(\alpha)$ where α is a root of $f(x) = x^6 + 2x^5 - 3x^3 + 2x - 1$. When we take $\alpha^{(0)} \approx 0.81$, the algorithm finds 983 distinct elements of which 662 are units. The highest norm we encounter is 211. As we can see in Figure 8, the value $\frac{L2-quality}{L2CK-constant}$ is approximately 2.5, hence the minimum of the L2-qualities we find lies close to $c(\boldsymbol{\alpha})$. When we start with the other real embedding, namely the one where $\alpha^{(0)} \approx -1.23$, this ratio $\frac{L2-quality}{L2CK-constant}$ is approximately 1619 and we see that the algorithm only detects 291 units out of the 999 distinct elements. Also the highest norm the algorithm finds is 79699.

To get some more insight how many elements of small norm the algorithm finds, we listed for each number field the value of $|\Omega_{F,a}|$, where we let a run over the 5 smallest integers for which $\Omega_{F,a} \neq \emptyset$. The results are listed in Table 4.

n=3, D=49:

root	$\lvert\Omega_{F,1}\rvert$	$\lvert\Omega_{F,7}\rvert$	$\lvert\Omega_{F,13}\rvert$
−1.80194	604	105	19
−0.44504	778	0	0
1.24698	807	0	0

n=4, D=275:

root	$\lvert\Omega_{F,1}\rvert$	$\lvert\Omega_{F,9}\rvert$	$\lvert\Omega_{F,11}\rvert$	$\lvert\Omega_{F,19}\rvert$	$\lvert\Omega_{F,25}\rvert$
1.68941	494	0	86	55	25
−1.68941	574	179	50	23	6

	root	$\lvert\Omega_{F,1}\rvert$	$\lvert\Omega_{F,11}\rvert$	$\lvert\Omega_{F,19}\rvert$	$\lvert\Omega_{F,25}\rvert$	$\lvert\Omega_{F,29}\rvert$
n=4, D=725:	3.38705	256	63	40	16	18
	−3.38705	326	86	55	25	20
	1.58993	515	106	69	37	13
	−1.58993	543	110	66	34	8

	root	$\lvert\Omega_{F,1}\rvert$	$\lvert\Omega_{F,11}\rvert$	$\lvert\Omega_{F,13}\rvert$	$\lvert\Omega_{F,17}\rvert$
For n=5, D=1609:	−1.68941	837	10	5	1

	root	$\lvert\Omega_{F,1}\rvert$	$\lvert\Omega_{F,11}\rvert$	$\lvert\Omega_{F,23}\rvert$	$\lvert\Omega_{F,43}\rvert$	$\lvert\Omega_{F,67}\rvert$
For n=5, D=14641:	−1.91899	152	25	74	50	34
	−1.30972	360	58	153	81	47
	−0.28463	498	70	193	69	35
	0.83083	517	72	159	84	34
	1.68251	207	35	99	74	52

	root	$\lvert\Omega_{F,1}\rvert$	$\lvert\Omega_{F,17}\rvert$	$\lvert\Omega_{F,19}\rvert$	$\lvert\Omega_{F,23}\rvert$	$\lvert\Omega_{F,25}\rvert$
For n=6, D=28037:	−1.23795	291	46	39	40	52
	0.80779	662	82	65	52	70

	root	$\lvert\Omega_{F,1}\rvert$	$\lvert\Omega_{F,29}\rvert$	$\lvert\Omega_{F,41}\rvert$	$\lvert\Omega_{F,49}\rvert$	$\lvert\Omega_{F,71}\rvert$
For n=6, D=300125:	−2.91560	8	3	2	1	4
	−0.77068	654	145	98	8	36
	−0.72009	655	142	87	10	32
	0.27505	289	173	106	17	86
	1.11366	177	102	65	13	42
	2.01766	69	7	8	2	2

Table 4: Number of elements of small norm.

We see that the value of a for which $\Omega_{F,a}$ is non-empty is independent of the choice of embedding, at least for small values of a (with the exception for the number field with $D = 275$, there we find that $\Omega_{F,9}$ is non-empty for the first, but empty for the second embedding). The cardinality of these sets does depend on the embedding. Also we see that when an embedding leads to a lot of units, also the other small norms occur more often. A remarkable fact is that we only encounter elements of which the norm is odd and this also holds for the elements with norms higher than displayed in these tables.

We have seen that the algorithm is capable of detecting small values of linear forms in integer points, even in high dimensions. When we pick $(\alpha_1, \ldots, \alpha_n)$ such that $(1, \alpha_1, \ldots, \alpha_n)$ is an integral basis of a number field of degree < 6 with small discriminant, the quality of the approximations lies close to the theoretical value $c(\boldsymbol{\alpha})$ given by Cusick and Krass, even though the LLL-algorithm is known to give suboptimal results. We also have seen that when we start the algorithm with the right choice for a real embedding, we can use it to detect elements of small norm.

References

[1] John WS Cassels. *An introduction to Diophantine approximation*, volume 1957. University Press Cambridge, 1957.

[2] TW Cusick and S Krass. Formulas for some diophantine approximation constants. *Journal of the Australian Mathematical Society (Series A)*, 44(03):311–323, 1988.

[3] AJ Brentjes and Multi-dimensional Continued Fraction Algorithms. Mathematical centre tracts. *Multi-dimensional continued fraction algorithms*, 145, 1981.

[4] JC Lagarias. Geodesic multidimensional continued fractions. *Proceedings of the London Mathematical Society*, 3(3):464–488, 1994.

[5] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and Lászl󠁯ó Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[6] Wieb Bosma and Ionica Smeets. Finding simultaneous diophantine approximations with prescribed quality. *The Open Book Series*, 1(1):167–185, 2013.

[7] Frits Beukers. Geodesic continued fractions and LLL. *Indagationes Mathematicae*, 25(4):632–645, 2014.

[8] H Davenport. On a theorem of furtwängler. *Journal of the London Mathematical Society*, 1(2):186–195, 1955.

[9] Tom M Apostol. *Modular functions and Dirichlet series in number theory*, volume 41. Springer Science & Business Media, 2012.

[10] JWS Cassels. *Rational quadratic forms, volume 13 of London Mathematical Society Monographs*. London Academic Press, 1978.

[11] Ph Furtwängler. Über die simultane approximation von irrationalzahlen. *Mathematische Annalen*, 96(1):169–175, 1927.

[12] HJ Godwin. Real quartic fields with small discriminant. *Journal of the London Mathematical Society*, 1(4):478–485, 1956.

[13] HJ Godwin. On quartic fields of signature one with small discriminant. ii. *Mathematics of Computation*, pages 707–711, 1984.

[14] A Schwarz, M Pohst, and F Diaz y Diaz. A table of quintic number fields. *mathematics of computation*, 63(207):361–376, 1994.

[15] Michael Pohst. On the computation of number fields of small discriminants including the minimum discriminants of sixth degree fields. *Journal of Number Theory*, 14(1):99–117, 1982.

A The mathematica notebook

The module `Start[v]` receives the vector of irrationals $v = \{\alpha_1, \ldots, \alpha_n\}$.

```
Start[vv_] :=
 Module[{qmaxSup, qualitiesSup, v, qminSup, qminL2, qmaxL2,
   qualitiesL2, stepcounter, norm, L2, Sup, np, nv, norms, precision,
   precisionValues, precisionBest},
  v = vv;
  PrependTo[v, 1];
  np = 700;
  nv = N[v, np];
  stepcounter = 0;

(*The following booleans define which values we
calculate at the end of each approximation step*)

  norm = False; (*Calculate the norm of the elements*)
  L2 = True; (*Calculate L2-quality*)
  Sup = True; (*Calculate sup-quality*)
  precision = True; (*Calculate precision*)

  A = Initialize[N[v, np]];
  InitialReduction[];

  While[! Complete,
   stepcounter++;

   ApproximationStep[];

   If[Accuracy[t] <= 0 || Accuracy[A] <= 0 || stepcounter > 999,
    Complete = True];

   If[L2,
    qualitiesL2 = ComputeQualityL2[nv];
    qminL2 = Min[Abs[qualitiesL2]];
    qmaxL2 = Max[Abs[qualitiesL2]],
    qualitiesL2 = {};
    qminL2 = {};
    qmaxL2 = {}];

   If[Sup,
    qualitiesSup = ComputeQualitySup[nv];
    qminSup = Min[Abs[qualitiesSup]];
    qmaxSup = Max[Abs[qualitiesSup]],
    qualitiesSup = {};
    qminSup = {};
    qmaxSup = {}];

   If[precision,
    precisionValues = ComputePrecision[nv];
    precisionBest = Min[Abs[precisionValues]],
    precisionBest = {}];
   If[norm,
    norms = ComputeNorm[v], norms = {}];


   AppendTo[
    table, {v, transform, stepcounter, shiftcounter, swapcounter,
     shiftAppStep, swapAppStep, Accuracy[t],
     Max[IntegerLength[Delete[transform, 1]]],
     Min[IntegerLength[Delete[transform, 1]]], qminL2, qmaxL2,
     qualitiesL2, qminSup, qmaxSup, qualitiesSup, norms,
     precisionBest}];
   ]]
```

In `Initialize[v]` we compute the initial values of B_{ij} and C_i. We can recall the value of B_{ij} with `B[[i,j,1]]*t+B[[i,j,2]]` and the value of C_i with `C[[i,1]]*t+C[[i,2]]`. This module returns the list $\{B, C\}$. Also we define some global variables which are used in several modules.

```
Initialize[vv_] := Module[{B, C, v},
  v = vv;
  n = Length[v];
  t = 1;
  w = 3/4;
  table = {{"v", "transform", "stepcounter", "shiftcounter",
     "swapcounter", "shiftAppStep", "swapAppStep", "Accuracy[t]",
     "Max[IntegerLength[transform]]", "Min[IntegerLength[transform]]",
      "qminL2", "qmaxL2", "qualitiesL2", "qminSup", "qmaxSup",
     "qualitiesSup", "norms", "precision"}};
  (*after each approximation step we store the information we in this
 table and we can transport this to Excel*)

  transform = IdentityMatrix[n];
(*this matrix keeps hold of the transformations and the
  columns of this matrix will give the approximations*)

  shiftcounter = 0; (*counts the total number of shifts*)
  swapcounter = 0; (*counts the total number of swaps*)
  swapAppStep = 0; (*counts the number of critical swaps*)
  shiftAppStep = 0;(*counts the number of critcal shifts*)
  stepcounter = 0; (*counts the number of approximation steps*)

  Complete = False; (*This boolean becomes true when the approximation process has to stop

  B = ConstantArray[0, {n, n, 2}];
  C = ConstantArray[0, {n - 1, 2}];
  For[i = 1, i <= n, i++,
   B[[i, i]] = {1, 0}];
  For[i = 2, i <= n, i++,
   B[[1, i]] = {v[[i]], 0}];
  For[i = 1, i <= n - 1, i++,
   C[[i]] = {1, 0}];
  C[[1]] = {v[[2]]^2, 1};

  Return[{B, C}]]
```

In `InitialReduction[]` we perform the first shifts to make sure that all mu_{ij} lie in the interval $[-0.5; 0.5]$, thus we check all the inequalties $2|B_{1j}| \leq B_{11}$ for $t = 1$.

```
InitialReduction[] := Module[{B11, B1j, mid, a},

  B11 = A[[1, 1, 1]][[1]]*t + A[[1, 1, 1]][[2]]; (*Computes value of B_11*)
  For[j = 2, j <= n, j++,
   B1j = A[[1, 1, j]][[1]]*t + A[[1, 1, j]][[2]];
   If[2*Abs[B1j] > B11, (*If the inequality is not met, we need to perform a shift*)
       mid = B1j/B11;(*We use mid to calculate with which value we have to shift*)
    a = Floor[0.5 - mid];
    Shift[1, j, a] (*We perform the shift x_1->x_1+a x_j*)
    ]
   ]
  ]
```

The module `ApproximationStep[]` performs one approximation step and calls for the corresponding critical shift or swap. It calls for `Reductionstep2[]` and `Reductionstep3[]` to make the new form LLL-reduced.

```
ApproximationStep[] := Module[{plan, a, i, j, mij},

  plan = MakePlan[];
  If[! Complete, (*While making the plan,
   Complete can become True in the ComputeInt module,
   then we skip the next part and the process stops*)

   If[plan[[1]], (*This means we have to shift*)
    shiftAppStep++;

    a = -1; (*a decides whether we shift with -1 of 1*)

    i = plan[[2]];
    j = plan[[3]];
    mij = (A[[1, i, j, 1]]*t + A[[1, i, j, 2]])/(A[[1, i, i, 1]]*t +
        A[[1, i, i, 2]]);

    If[mij < 0, a = 1];
    Shift[i, j, a];
    If[j == i + 1, ReductionStep2[]];
    ReductionStep3[];(*For partial reduction we can skip this step*)
    , (*else we have to swap*)
    swapAppStep++;

    Swap[plan[[2]]];

    ReductionStep2[];
    ReductionStep3[] (*For partial reduction we can skip this step*)
    ]
  ]]
```

The module `MakePlan[]` uses the list of intervals which is the output of `ComputeInterval[]`. We calculate the next value of t and return a plan. A plan is a list of the form $shift, i, j$, where shift is a boolean. If shift is true we need to perform a shift, otherwise a swap. The i and j tell us which shift or swap we have to perform.

```
MakePlan[] := Module[{int, shift, i, j, upperbound},


  int = ComputeInterval[];
  shift = True;
  upperbound = Max[int[[1, 3]]];
  i = int[[1, 1]];
  j = int[[1, 2]];
  For[k = 1, k <= Length[int], k++,
   If[Max[int[[k, 3]]] < upperbound,
    upperbound = Max[int[[k, 3]]];
    i = int[[k, 1]];
    j = int[[k, 2]]]];
  If[j == -1, shift = False];
  t = upperbound; (*assigns the new value for t*)

  Return[{shift, i, j}]]
```

The module `ComputeInterval[]` computes for each inequality the intervals for t for which the inequality is met. It returns an array with elements of the form $\{int, i, j\}$ where int is the interval and i and j denote the corresponding inequality. If $j = -1$ the interval belongs to the inequality $\omega B_{ii} \leq C_i$ is violated (so we have to swap), otherwise the interval belongs to the inequality $2|B_{ij}| < B_{ii}$ (so we have to shift).

```
ComputeInterval[] := Module[{Int, Bij, Bii, int1, int2, int, Ci},

  Int = {};

  (*In the following For-loop we check the inequalities 2|Bij|<Bii*)
  For[i = 1, i <= n, i++,
   For[j = i + 1, j <= n, j++,
    Bij = 2 A[[1, i, j]];
    Bii = A[[1, i, i]];
    If[Bij[[1]] - Bii[[1]] == 0 || -Bii[[1]] - Bij[[1]] == 0,
     Complete = True]; (*If one of these is zero,
    we can not make a new plan and we stop the approximation.
    This can happen when two elements of the input vector are
dependent over the integers*)

    If[! Complete, (*We have to check both 2Bij<Bii and -Bii<2Bij.
     Then we compute the intersection of these.*)
     If[Bij[[1]] - Bii[[1]] > 0,
      int1 =
       Interval[{0, (Bii[[2]] - Bij[[2]])/(Bij[[1]] - Bii[[1]])}],
      int1 =
       Interval[{(Bii[[2]] - Bij[[2]])/(Bij[[1]] - Bii[[1]]),
         Infinity}]];
     If[-Bii[[1]] - Bij[[1]] > 0,
      int2 =
       Interval[{0, (Bij[[2]] + Bii[[2]])/(-Bii[[1]] - Bij[[1]])}],
      int2 =
       Interval[{(Bij[[2]] + Bii[[2]])/(-Bii[[1]] - Bij[[1]]),
         Infinity}]];
     int = IntervalIntersection[int1, int2];

     AppendTo[Int, {i, j, int}]
     ]];

   (*In the following For-
   loop we check the inqualities wBii <= Ci*)
   For[i = 1, i <= n - 1, i++,
    Bii = w*A[[1, i, i]];
    Ci = A[[2, i]];
    If[Bii[[1]] - Ci[[1]] > 0,
     int = Interval[{0, (Ci[[2]] - Bii[[2]])/(Bii[[1]] - Ci[[1]])}],
     int =
      Interval[{(Ci[[2]] - Bii[[2]])/(Bii[[1]] - Ci[[1]]), Infinity}]];
    AppendTo[
     Int, {i, -1, int}]]]; (*The -1 indicates that we have to perform a swap*)
  Return[Int]]
```

The module `Shift[]` performs a shift, hence it updates the transformation matrix and the determinants.

```
Shift[rr_, ss_, aa_] := Module[{r, s, a, B, C},
  (*This performs the shift x_r\[Rule]x_r+ax_s *)
  r = rr;
  s = ss;
  a = aa;

  shiftcounter++;

  (*Update transformmatrix*)
  For[i = 1, i <= n, i++,
   transform[[i, s]] += a transform[[i, r]]];

  (*Update determinants*)
  If[r == s - 1,
   A[[2, r]] += 2 a A[[1, r, s]] + a^2 A[[1, r, r]]];
  For[i = 1, i <= r, i++,
   A[[1, i, s]] += a*A[[1, i, r]]]

  ]
```

The module `Swap[]` performs a swap, hence it updates the transformation matrix and the determinants.

```
Swap[rr_] :=
 Module[{r, old, Br1r1, Br2r2, Brijb, Bjj1, Bjj, mjj1, shift,
   oldtrans},

  r = rr;

  swapcounter++;

  (*Update the transformmatrix*)
  For[i = 1, i <= n, i++,
   oldtrans = transform[[i, r]];
   transform[[i, r]] = transform[[i, r + 1]];
   transform[[i, r + 1]] = oldtrans;
   ];
  (*Update determinants*)

  (*update rule 4 and 5*)
  For[j = r + 2, j <= n, j++,
   old = A[[1, r, j]];
   If[r == 1, Br1r1 = {0, 1}, Br1r1 = A[[1, r - 1, r - 1]]];

   If[A[[1, r, r, 2]] == 0,
    A[[1, r, j, 2]] =
     Round[(A[[1, r, r + 1, 2]]*A[[1, r, j, 1]] +
         A[[1, r, r + 1, 1]]*A[[1, r, j, 2]] +
         Br1r1[[2]]*A[[1, r + 1, j, 1]] +
         Br1r1[[1]]*A[[1, r + 1, j, 2]])/A[[1, r, r, 1]]];
    A[[1, r, j,
       1]] = (A[[1, r, r + 1, 1]]*A[[1, r, j, 1]] +
        Br1r1[[1]]*A[[1, r + 1, j, 1]])/(A[[1, r, r, 1]]);
    A[[1, r + 1, j, 2]] =
     Round[(A[[1, r + 1, r + 1, 2]]*old[[1]] +
         A[[1, r + 1, r + 1, 1]]*old[[2]] -
         A[[1, r, r + 1, 2]]*A[[1, r + 1, j, 1]] -
         A[[1, r, r + 1, 1]]*A[[1, r + 1, j, 2]])/(A[[1, r, r, 1]])];
    A[[1, r + 1, j,
       1]] = (A[[1, r + 1, r + 1, 1]]*old[[1]] -
        A[[1, r, r + 1, 1]]*A[[1, r + 1, j, 1]])/(A[[1, r, r, 1]]);,
    (*else*)
    A[[1, r, j, 2]] =
     Round[(A[[1, r, r + 1, 2]]*A[[1, r, j, 2]] +
         Br1r1[[2]]*A[[1, r + 1, j, 2]])/A[[1, r, r, 2]]];
    A[[1, r, j,
       1]] = (A[[1, r, r + 1, 2]]*A[[1, r, j, 1]] +
        A[[1, r, r + 1, 1]]*old[[2]] +
        Br1r1[[2]]*A[[1, r + 1, j, 1]] +
        Br1r1[[1]]*A[[1, r + 1, j, 2]] -
        A[[1, r, j, 2]]*A[[1, r, r, 1]])/A[[1, r, r, 2]];
    Brijb = A[[1, r + 1, j, 2]];
    A[[1, r + 1, j, 2]] =
     Round[(A[[1, r + 1, r + 1, 2]]*old[[2]] -
         A[[1, r, r + 1, 2]]*A[[1, r + 1, j, 2]])/A[[1, r, r, 2]]];
    A[[1, r + 1, j,
       1]] = (A[[1, r + 1, r + 1, 2]]*old[[1]] +
      A[[1, r + 1, r + 1, 1]]*old[[2]] -
      A[[1, r, r + 1, 2]]*A[[1, r + 1, j, 1]] -
      A[[1, r, r + 1, 1]]*Brijb -
      A[[1, r + 1, j, 2]]*A[[1, r, r, 1]])/A[[1, r, r, 2]];]];
(*update rule 8*)

If[r > 1,

 If[r == 2, Br2r2 = {0, 1}, Br2r2 = A[[1, r - 2, r - 2]]];
 If[A[[1, r - 1, r - 1, 2]] == 0,
  A[[2, r - 1, 2]] =
   Round[(Br2r2[[2]]*A[[2, r, 1]] + Br2r2[[1]]*A[[2, r, 2]] +
       2*A[[1, r - 1, r + 1, 2]]*A[[1, r - 1, r + 1, 1]])/
     A[[1, r - 1, r - 1, 1]]];
  A[[2, r - 1,
     1]] = (Br2r2[[1]]*A[[2, r, 1]] +
      A[[1, r - 1, r + 1, 1]]*A[[1, r - 1, r + 1, 1]])/
    A[[1, r - 1, r - 1, 1]];,(*else*)

  A[[2, r - 1, 2]] =
   Round[(Br2r2[[2]]*A[[2, r, 2]] +
       A[[1, r - 1, r + 1, 2]]*A[[1, r - 1, r + 1, 2]])/
     A[[1, r - 1, r - 1, 2]]];

  A[[2, r - 1,
     1]] = (Br2r2[[2]]*A[[2, r, 1]] + Br2r2[[1]]*A[[2, r, 2]] +
      2*A[[1, r - 1, r + 1, 2]]*A[[1, r - 1, r + 1, 1]] -
      A[[2, r - 1, 2]]*A[[1, r - 1, r - 1, 1]])/
    A[[1, r - 1, r - 1, 2]];
  ]];
(*update rule 2 en 3*)

For[i = 1, i < r, i++,
 old = A[[1, i, r]];
 A[[1, i, r]] = A[[1, i, r + 1]];
 A[[1, i, r + 1]] = old];

(*update rule 9*)

If[r < n - 1,
 If[A[[1, r + 1, r + 1, 2]] == 0,
   A[[2, r + 1, 2]] =
    Round[(A[[1, r + 2, r + 2, 2]]*A[[2, r, 1]] +
        A[[1, r + 2, r + 2, 1]]*A[[2, r, 2]] +
        2*A[[1, r + 1, r + 2, 2]]*A[[1, r + 1, r + 2, 1]])/
      A[[1, r + 1, r + 1, 1]]];
   A[[2, r + 1,
      1]] = (A[[1, r + 2, r + 2, 1]]*A[[2, r, 1]] +
       A[[1, r + 1, r + 2, 1]]*A[[1, r + 1, r + 2, 1]])/(A[[1,
        r + 1, r + 1, 1]]);

   ,(*else*)
   A[[2, r + 1, 2]] =
    Round[(A[[1, r + 2, r + 2, 2]]*A[[2, r, 2]] +
          A[[1, r + 1, r + 2, 2]]*A[[1, r + 1, r + 2, 2]])/
        A[[1, r + 1, r + 1, 2]]];
     A[[2, r + 1,
        1]] = (A[[1, r + 2, r + 2, 2]]*A[[2, r, 1]] +
         A[[1, r + 2, r + 2, 1]]*A[[2, r, 2]] +
         2*A[[1, r + 1, r + 2, 2]]*A[[1, r + 1, r + 2, 1]] -
         A[[2, r + 1, 2]]*A[[1, r + 1, r + 1, 1]])/
       A[[1, r + 1, r + 1, 2]]];

   ];
  (*update rule 1 en 7*)

   old = A[[1, r, r]];
  A[[1, r, r]] = A[[2, r]];
  A[[2, r]] = old;

  (*After a swap, we always need to check for possible shifts,
  hence it is included in the Swap method.*)

  For[j = (r - 1), j <= (r + 1), j++,
   If[(j < 1 || j > n - 1), Continue[]];

   Bjj1 = A[[1, j, j + 1, 1]]*t + A[[1, j, j + 1, 2]];
   Bjj = A[[1, j, j, 1]]*t + A[[1, j, j, 2]];

   If[2*Abs[Bjj1] > Bjj,
    mjj1 = Bjj1/Bjj;
    shift = Floor[0.5 - mjj1];
    Shift[j, j + 1, shift];
    ]]
  ]
```

After computing the new t and performing the corresponding critical shift or swap, we sometimes have to perform more shifts and swaps to make the new form LLL-reduced. Reductionstep2[] executes the required swaps and Reductionstep3[] executes the required shifts.

```
ReductionStep2[] := Module[{Bii, Ci},

  For[i = 1, i < n, i++,
   Bii = A[[1, i, i, 1]]*t + A[[1, i, i, 2]];
   Ci = A[[2, i, 1]]*t + A[[2, i, 2]];
   If[w *Bii > Ci,
    Swap[i];
    i = 0;]]]

ReductionStep3[] := Module[{Bij, Bii, mij, shift},

  For[j = n, j > 1, j--,
   For[i = j - 1, i > 0, i--,
    Bij = A[[1, i, j, 1]]*t + A[[1, i, j, 2]];
    Bii = A[[1, i, i, 1]]*t + A[[1, i, i, 2]];
    If[2*Abs[Bij] > Bii,
      mij = Bij/Bii;
      shift = Floor[0.5 - mij];
      Shift[i, j, shift]
     ]]]
  ]
```

The module ComputeQualityL2[v] computes for each column of the transformation matrix the value

$$|q+p_1\alpha_1+\ldots+p_n\alpha_n|\cdot||\boldsymbol{p}||_2^n.$$

It returns a list with all these values.

```
ComputeQualityL2[v_] :=
 Module[{vv, testvalue, value, values, list, combi, L2, norm},
  list = {};
  vv = v;

  For[i = 1, i <= n, i++,
   combi = transform[[All,i]];
   L2 = 0;
   For[k = 2, k <= n, k++,
    L2 += combi[[k]]^2];
   testvalue = L2^((n - 1)/2);
   value = vv.combi;
   quality = N[value*testvalue, 20];
   AppendTo[list, quality]
   ];
  Return[list]
  ]
```

The module ComputeQualitySup[v] computes for each column of the transformation matrix the value

$$|q+p_1\alpha_1+\ldots+p_n\alpha_n|\cdot||\boldsymbol{p}||_\infty^n.$$

It returns a list with all these values.

```
ComputeQualitySup[v_] :=
 Module[{vv, testvalue, value, values, list, combi, max, n, norm},

  list = {};
  vv = v;


  For[i = 1, i <= Length[transform], i++,
   combi =
    transform[[All,
      i]];
   max = Max[Abs[Delete[combi, {1}]]];
   testvalue = max^(Length[vv] - 1);
   value = vv.combi;
   quality = N[value*testvalue, 40];
   AppendTo[list, quality];
   ];
  Return[list]
  ]
```

The module `ComputePrecision[v]` computes for each column of the transformation matrix the value

$$|q + p_1\alpha_1 + \ldots + p_n\alpha_n|.$$

It returns a list with all these values.

```
ComputePrecision[v_] :=
 Module[{vv, testvalue, value, values, list, combi, L2, norm,
   precision},

  list = {};
  vv = v;

  For[i = 1, i <= n, i++,
   combi =
    transform[[All, i]];
   value = vv.combi;
   precision = N[value, 200];
   AppendTo[list, precision]
   ];
  Return[list]
  ]
```

When the elements of the input vector are algebraic integers, `ComputeNorm[v]` computes for each column of the transformation matrix the norm of the corresponding element. It returns a list with all these values.

```
ComputeNorm[v_] := Module[{combi, list, vv},

  list = {};
  For[i = 1, i <= Length[transform], i++,
   combi = transform[[All, i]];
   vv = v;
   AppendTo[list, AlgebraicNumberNorm[combi.vv]]];
  Return[list]]
```