Chapter 7

Solving Systems of Diophantine Linear Equations

7

7 Solving Systems of Diophantine Linear Equations

7 Solving Systems of Diophantine Linear Equations

In this chapter we consider systems of linear equations whose solutions are restricted to the integers. Linear equations of this form are called *Diophantine* linear equations. In Chapter 8 we will reduce the problem of finding linear codes with prescribed minimum distance to solving systems of Diophantine linear equations. If we try to solve these systems it is crucial to have fast methods at hand. Here, we study one possible approach based on so called lattice basis reduction. In Section 1.8 we saw an algorithm for determining the minimum distance of a linear code. Section 7.8 contains another minimum distance algorithm, also based on lattice basis reduction.

With Gaussian elimination we are able to solve linear systems $A \cdot x = d$ of equations for vectors $x \in \mathbb{R}^n$ easily[1]. The same algorithm works also if we restrict to $x \in \mathbb{Q}^n$. Then, since we can multiply the whole system with the least common multiple of all denominators, we can also solve these systems over $\mathbb{Z}$. Unfortunately, the size of the denominators can grow very rapidly. So, Gaussian Elimination does not longer run in polynomial time. But there exist algorithms to compute the Hermitian normal form (HNF) efficiently, i.e. in polynomial time, see for example [39]. Thus, with the help of the HNF we can solve systems of Diophantine linear equations easily.

The situation changes when we have to solve systems of linear *inequalities* over the integers or, equivalently, if we have to find nonnegative integral solutions of systems of linear equations. Equally hard problems arise if the variables x_i are restricted to integers from intervals $l_i \leq x_i \leq r_i$ for $i \in n$. The problem to decide if there is such a vector x is known to be NP-complete. At present, no algorithm is known which decides in a number of steps that is polynomial in the size of the input for this problem if there is a solution or not. Here, we restrict our attention to Diophantine linear equations of the following form.

$$A \cdot x = d, \quad l \leq x \leq r,$$

for given $A \in \mathbb{Z}^{m \times n}, d \in \mathbb{Z}^m, l, r \in \mathbb{Q}^n$, where $l \leq x \leq r$ means $l_i \leq x_i \leq r_i$, for each $i \in n$. We ask for solutions $x \in \mathbb{Z}^n$ with $l \leq x \leq r$.

Example In Chapter 8, Example 8.4.4, the following system of Diophantine linear equations occurs during the construction of linear codes with prescribed **7.0.1**

[1] for technical reasons we use the column convention in the present chapter

minimum distance:

$$\begin{pmatrix} 2 & 2 & 0 & -1 & 0 & 0 \\ 1 & 1 & 2 & 0 & -1 & 0 \\ 0 & 3 & 1 & 0 & 0 & -1 \\ 3 & 6 & 4 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 14 \end{pmatrix},$$

where $x_0 \in \{0,1,2,3,4\}$, $x_1 \in \{0,1,2\}$, $x_2 \in \{0,1,2,3\}$ and $x_i \in \{0,1,2,3,4,5\}$ for $i \in \{3,4,5\}$. It is easy to check that $x = (0,1,2,2,5,5)^\top$ is an integer solution of the system of equations which also satisfies the additional lower and upper bounds. ◇

Several equally hard variations of this problem exist. The knapsack problem and the subset sum problem are just two instances.

- The *knapsack problem:* Given nonnegative integers c_i, w_i, $i \in n$, and k, find a subset $S \subseteq \{0,1,\ldots,n-1\}$ such that $\sum_{j\in S} w_j \leq k$ and $\sum_{j\in S} c_j$ is maximal.
- The *subset sum problem:* Given nonnegative numbers w_i, $i \in n$, and k, find a subset $S \subseteq \{0,1,\ldots,n-1\}$ such that $\sum_{j\in S} w_j = k$.

7.0.2 **Example** Let $w = (31,41,59,26,53,58,97,93,23,84,62)$, $k = 314$ and $c = (1,1,1,1,1,1,1,1,1,1,1)$.

- The subset sum problem asks for subsets $S \subseteq \{0,1,2,\ldots,10\}$ such that $\sum_{i\in S} w_i = 314$. There are three solutions: $\{3,4,5,7,9\}$, $\{0,3,4,5,9,10\}$, and $\{0,2,3,6,8,9\}$.
- In the knapsack problem, we ask for subsets $S \subseteq \{0,1,\ldots,10\}$ of maximal size subject to the condition that $\sum_{i\in S} w_i \leq 314$. Here, the solution is $S = \{0,2,3,4,5,8,10\}$, $\sum_{i\in S} c_i = 7$ and $\sum_{i\in S} w_i = 312 \leq 314$. ◇

As we will see in Section 7.8, the problem of computing the minimum distance of certain linear codes can be reduced to a problem of solving a Diophantine system of linear equations. Also, in Chapter 8, we will use systems of Diophantine linear equations to construct optimal codes. Many further objects from Discrete Mathematics can be constructed in a similar fashion. In fact, Combinatorial Designs, Steiner systems and covering codes have all been constructed in a similar way by means of Diophantine equations.

Many algorithms for solving these problems have been proposed. Some of them rely on relaxation techniques and use Linear Programming. Other approaches use backtracking. The approach used here is based on lattices[2] and

[2] in this chapter lattices are geometrical objects, different from the definition in 3.2.24

on a very important method invented by Lenstra, Lenstra and Lovász [125] – the celebrated LLL-algorithm. This method has been applied successfully to break certain cryptosystems based on the knapsack problem [119].

The first step is to transform the problem of finding the solutions of linear Diophantine equation systems into a problem involving lattices. A lattice is just the set of integer linear combinations of a given set of linearly independent vectors in a real vector space. In this setting, the problem can be reduced to the question of finding sufficiently short vectors in a suitable lattice. Here, short is usually meant in connection to a norm, like the ℓ_∞-norm or the Euclidean norm. To find these short vectors in polynomial time, we apply the LLL-algorithm. In a second step, we use exhaustive enumeration to find *all* vectors which are solutions of our original problem. This last step needs exponential time.

With this approach many finite incidence structures could be constructed, see [15], [16], [25], [28], [29], [199] and the literature cited there.

7.1 Lattices

Let us recall briefly the basic definitions and fundamental theorems of the theory of lattice. For a thorough introduction into the subject we refer the reader to [77], for instance.

- As usual, let $\mathbb{R}^n$ denote the real Euclidean n-dimensional space. Its elements $v \in \mathbb{R}^n$ are written as column vectors $v = (v_0, v_1, \ldots, v_{n-1})^\top$.
- For $q \in \mathbb{R}$, $q \geq 1$, we define the *ℓ_q-norm* by

$$\|-\|_q \,:\, \mathbb{R}^n \to \mathbb{R} \,:\, v \mapsto \left(\sum_{i\in n} |v_i|^q\right)^{1/q},$$

  and the *ℓ_∞-norm* as follows:

$$\|-\|_\infty \,:\, \mathbb{R}^n \to \mathbb{R} \,:\, v \mapsto \max_{i\in n} |v_i|\,.$$

- For $m \in \mathbb{N}$, the vectors $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)} \in \mathbb{R}^n$ span a subspace of $\mathbb{R}^n$ which we denote by

$$\langle b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}\rangle := \left\{\sum_{i\in m} x_i b^{(i)} \mid x_i \in \mathbb{R}\,,\ i \in m\right\}.$$

The notation for a subspace $\langle b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}\rangle$ is not to be confused with the standard bilinear form

$$\langle v, w\rangle = \sum_{i\in n} v_i \cdot w_i$$

for $v, w \in \mathbb{R}^n$. But the meaning should be clear from the context.

The basic notions are the following ones:

7.1.1 **Definition (lattice)** Let $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ be m linearly independent vectors in $\mathbb{R}^n$.

- The set
$$L(b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}) := \left\{ \sum_{i \in m} u_i \cdot b^{(i)} \;\middle|\; u_i \in \mathbb{Z},\, i \in m \right\} \subset \mathbb{R}^n$$
is called the *lattice (of vectors)* with *basis* $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$.
- The *rank* m of a lattice L with basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ is the dimension of the $\mathbb{R}$-subspace $\langle b^{(0)}, b^{(1)}, \ldots, b^{(m-1)} \rangle$ which is spanned by the basis.
- We will write
$$B := \left(b^{(0)} \mid \ldots \mid b^{(m-1)} \right)$$
for the $n \times m$-matrix whose columns are the vectors $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$. If $L = L(b^{(0)}, b^{(1)}, \ldots, b^{(m-1)})$, then B is called a *generator matrix* of L. ◇

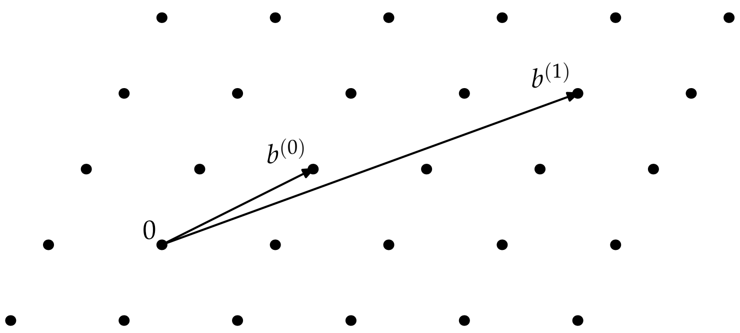


Fig. 7.1 A rank 2 lattice spanned by $b^{(0)}$ and $b^{(1)}$

It is well known [77, p. 18] that a lattice of vectors in $\mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$.

For a lattice $L \subset \mathbb{R}^n$, the most important (and difficult) algorithmic problems can be described as follows.

7.1.2 **Algorithmic problems for a given lattice L**

- The *shortest vector problem* (SVP): Find an ℓ_q-shortest vector in L, i.e. find an element w in L such that
$$\|w\|_q = \min\{\|w'\|_q \mid w' \in L \setminus \{0\}\}.$$

This question is most interesting for the Euclidean norm, the ℓ_1-norm, and the ℓ_∞-norm.

- The *closest vector problem* (CVP): Given a vector $v \in \mathbb{R}^n$ find a lattice vector w which is closest to v in the ℓ_q-norm, i.e. such that
$$\|v - w\|_q = \min\{\, \|v - w'\|_q \mid w' \in L\}\,.$$
- The *lattice basis reduction*: Given a basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ of the lattice L compute a new basis $b'^{(0)}, b'^{(1)}, \ldots, b'^{(m-1)}$ of L consisting of "shortest" vectors. Here, the meaning of short will have to be made precise. ◇

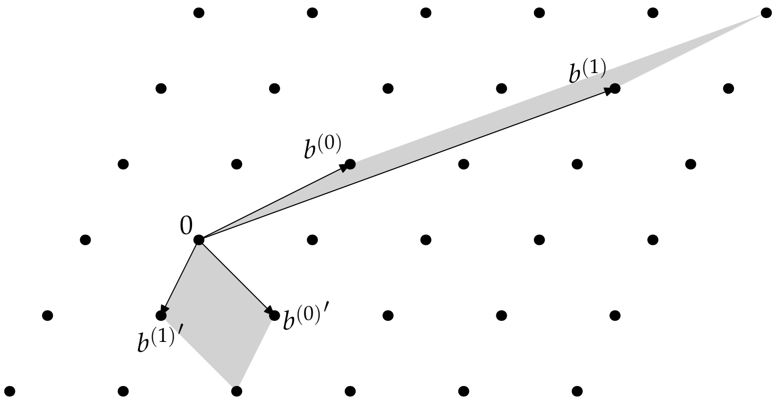


Fig. 7.2 Two different bases for $b^{(0)}, b^{(1)}$ and $b^{(0)\prime}, b^{(1)\prime}$ of the same lattice

For an overview on the algorithmic complexity of the above problems we refer to [147] and [199] and the literature cited there.

Concerning the last of the mentioned problems, we remark that the problem of finding a basis consisting of shortest vectors is not exactly defined provided the dimension is at least three. In fact, many different versions of the concept of a shortest basis exist. Two classical concepts are the reduced bases in the sense of Minkowski [150] and the reduced quadratic forms in the sense of Korkine and Zolotarev [113]. The latter aims at minimizing the orthogonality defect of a lattice basis, a concept which we will encounter in Section 7.5. Recently, one further variant has gained interest. In this, one finds a lattice basis minimizing the maximal length of any of its members, see [2], [23].

The above reduction concepts rely on the computation of shortest lattice vectors in sublattices and related lattices. Therefore, the problem of computing a reduced lattice basis in the sense of Minkowski or Korkine and Zolotarev is at least as hard as the shortest vector problem.

7.2 7.2 Diophantine Equations and Lattices

Subset sum problems. Lagarias and Odlyzko [119] have introduced the techniques of lattices and lattice basis reduction to the solution of subset sum problems. Recall that these problems can be written as finding all solutions $x \in \{0,1\}^n$ of the system

7.2.1 $$A \cdot x = d$$

where A is a $1 \times n$ matrix over the integers and d is some integer. In fact, they introduced the lattice whose generator matrix is the $(m+n) \times (n+1)$-matrix

7.2.2 $$B := \left(\begin{array}{c|c} N \cdot (-d) & N \cdot A \\ \hline 0 & \\ \vdots & I_n \\ 0 & \end{array}\right)$$

where I_n denotes the identity matrix in $\mathbb{Z}^{n \times n}$ and N is a large integer constant. Let us call this lattice the Lagarias-Odlyzko lattice. It turns out that the solutions x of 7.2.1 are in bijection to certain short elements of the Lagarias-Odlyzko lattice. Namely, if $v = B \cdot w$ is an element of the lattice which is zero in the first m entries and where w is a $\{0,1\}$-vector whose first component is equal to one, then $(w_1, \ldots, w_n)^\top$ is a solution of the Diophantine system 7.2.1 and vice-versa. Moreover, the first m components of v are zero, and hence no entry of v is a nonzero multiple of the large integer constant N. This means that v is short in the Lagarias-Odlyzko lattice. Therefore, we see that solutions of 7.2.1 are short vectors in the Lagarias-Odlyzko lattice. This means that it is useful to attack this kind of Diophantine problem with the method of finding short vectors in lattices. We illustrate this by an example.

7.2.3 **Example** Consider the subset sum problem of 7.0.2. Setting $N = 100$, the generator matrix B of 7.2.2 of the Lagarias-Odlyzko lattice is

$$\left(\begin{array}{c|ccccccccccc}
-31400 & 3100 & 4100 & 5900 & 2600 & 5300 & 5800 & 9700 & 9300 & 2300 & 8400 & 6200 \\ \hline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right).$$

Multiplying this matrix by the vector $w = (1,0,0,0,1,1,1,0,1,0,1,0)^\top$ gives the vector $v = (0,0,0,0,1,1,1,0,1,0,1,0)^\top$. Therefore, $(w_1, w_2, \ldots, w_n)^\top$ is a solution to the subset sum problem: We have $w_i = 1$ if and only if $i \in S = \{4, 5, 6, 8, 10\}$. This set S solves the subset sum problem 7.0.2. Note that $\|v\|_2 = \sqrt{5}$ and $\|v\|_\infty = 1$.

Multiplying the matrix by the vector $w = (1, 1, 0, 1, 0, 2, 1, 1, -1, 1, 1, 0)^\top$ gives $v = (36500, 1, 0, 1, 0, 2, 1, 1, -1, 1, 1, 0)$. This means that $(w_1, w_2, \ldots, w_n)^\top$ does not solve the subset sum problem: The entries in $(w_1, w_2, \ldots, w_n)^\top$ are not all elements of $\{0, 1\}$. Furthermore, the linear equation $A \cdot (w_1, \ldots, w_n)^\top = d$ is violated. We note that $\|v\|_2 = \sqrt{1\,332\,250\,011}$ and $\|v\|_\infty = 36\,500$. ◇

Below, we will employ the following strategy. We will start with the lattice basis which is given by the columns of the generator matrix B. We will then compute another basis for the same lattice which consists of short vectors. This transformation from one lattice basis to another is known as *lattice basis reduction*. A very important algorithm to achieve this transformation is the LLL-algorithm which we will discuss in Section 7.6.

In the context of the subset sum problem and the Lagarias-Odlyzko lattice, one hopes that through the process of lattice basis reduction one will eventually arrive at vectors $v \in \mathbb{Z}^{n+m}$ which are of the form $v_i = 0$ for $i \in m$ and either $v_i \in \{0, 1\}$ for $m \leq i < m + n$ or, alternatively, $v_i \in \{0, -1\}$ for $m \leq i < m + n$. It is proved in [119] that for a large class of subset sum problems a solution will correspond to the shortest vector of the lattice 7.2.2.

The Euclidean norm of such vectors is bounded above by $\sqrt{n}$. But not every short vector is a solution. Since the Euclidean distance of a vector does not distinguish between entries $+1$ and -1, it may happen that short vectors are computed whose entries are 0 or ± 1. In fact, the "mixed sign case" happens frequently among the vectors $v \in L$ with $\|v\|_\infty = 1$.

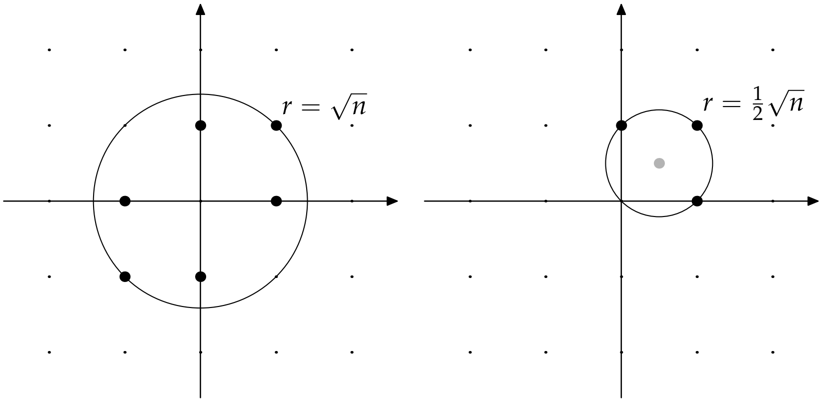


Fig. 7.3 Solution vectors for the lattice 7.2.2 (left) and for the lattice 7.2.4 (right) without the first component which is equal to zero

We can do better by appealing to the closest vector problem. The goal is to eliminate entries in the short lattice vectors which are -1. To do this, we introduce the vector $z = (0, \frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})^\top$, which is *not* contained in the Lagarias-Odlyzko lattice. We are now looking for the vectors closest to z in the Lagarias-Odlyzko lattice. In fact, since we are looking for $\{0,1\}$-vectors, we may restrict our search to lattice vectors v at distance

$$\|v - z\| = \sqrt{\sum_{i \in n} (v_i - z_i)^2} = \sqrt{\sum_{i \in n} 1/4} = 1/2\sqrt{n}.$$

The situation for $n = 3$ is illustrated in Fig. 7.3. The first component of the lattice vectors is not shown as it is zero. The black dots indicate lattice points, leading to solutions, which are either short (left picture) or close to z (right picture). To solve the closest vector type problem, we use the augmented and embedded Lagarias-Odlyzko lattice, generated by the $(m+n+1) \times (n+1)$-matrix

7.2.4
$$B := \left(\begin{array}{c|ccc} -N \cdot d & & N \cdot A & \\ \hline -1/2 & & & \\ \vdots & & I_n & \\ -1/2 & & & \\ \hline 1 & 0 & \cdots & 0 \end{array} \right).$$

This means, we add a zero component to the original Lagarias-Odlyzko lattice and add the new basis vector $(-Nd, -\frac{1}{2}, \ldots, -\frac{1}{2}, 1)^\top$. The last component ensures that the columns of 7.2.4 are linearly independent. Also, it serves as a bookkeeping device. Namely, it keeps track of whether the new basis vector was used in the expression of a lattice element in terms of the new basis. As before, $N \in \mathbb{N}$ is a large integer constant. The solutions of the subset sum problem now correspond to elements $v = B \cdot w \in \mathbb{Z}^{m+n+1}$ of the new lattice 7.2.4 where $v_i = 0$ for $i \in m$, $v_i \in \{-1/2, 1/2\}$ for $m \leq i < n+m$ and $|v_{n+m}| = |w_0| = 1$. For these vectors the maximum norm is equal to 1, and all lattice vectors which are not solutions of the subset sum problem have maximum norm greater than 1.

7.2.5 **Example** Consider again the subset sum problem from 7.0.2 and put $N = 100$. The extended Lagarias-Odlyzko lattice is generated by the matrix

$$\left(\begin{array}{c|ccccccccccc}
-31400 & 3100 & 4100 & 5900 & 2600 & 5300 & 5800 & 9700 & 9300 & 2300 & 8400 & 6200 \\
\hline
-1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right) .$$

Note that we have scaled all rows except for the first and the last one by a factor of 2 to clear denominators.

Multiplying this matrix from the right by the same vector $w = (1,0,0,0,1,1,1,0,1,0,1,0)^\top$ as in Example 7.2.3 produces the short lattice vector $v = (0,-1,-1,-1,1,1,1,-1,1,-1,1,-1,1)^\top$. Comparing this vector to the vector v of Example 7.2.3 shows that apart from the first entry we have replaced all zeros by -1s. Also, we have left in place the 1s and we have added a final entry. Furthermore, note that $\|v\|_2 = \sqrt{12}$ and $\|v\|_\infty = 1$. If the last component of v is equal to 1, then $v_i = -1$ corresponds to $i \notin S$, $v_i = 1$ corresponds to $i \in S$. If the last component of v is equal to -1 it is the other way round. $\diamond$

As shown in [43], this improvement enlarges the class of subset sum problems whose solutions are shortest vectors in the original Lagarias-Odlyzko lattice 7.2.4 enormously.

Systems of Diophantine linear equations. In order to solve the problem $A \cdot x = d$ for $A \in \mathbb{Q}^{m \times n}$ and $d \in \mathbb{Q}^m$ with $l \leq x \leq r$ for arbitrary bounds $l, r \in \mathbb{Q}^n$, our algorithm proceeds in two steps.

First, we compute a basis consisting of integer vectors $b^{(0)}, b^{(1)}, \ldots, b^{(n-m)}$ of the augmented system 7.2.6

$$\underbrace{\left(\begin{array}{c|c} \\ -d & \quad A \quad \\ \\ \end{array}\right)}_{=:\,A'} \cdot \begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix} = 0 \,. \qquad \mathbf{7.2.6}$$

In this system, the negative of the right hand side has been added to the coefficient matrix A on the left, to form the extended coefficient matrix A'. Correspondingly, a component x_0 has been added to the vector x.

Since we can assume that the augmented matrix A' has full row-rank m, the kernel of the system 7.2.6 has dimension $n - m + 1$. Of course, only solutions of 7.2.6 with $x_0 = 1$ are interesting. Several polynomial-time algorithms are known to compute the integer basis of this kernel in $\mathbb{Z}^{n+1}$, as described in [39]. Since it is desirable for the second step of our algorithm to have a basis $b^{(0)}$, $b^{(1)}, \ldots, b^{(n-m)} \in \mathbb{Z}^{n+1}$ consisting of short vectors, algorithms based on lattice basis reduction are preferred [39], [198].

In order to handle the lower bounds, we reformulate the problem in such a way that the lower bounds on the variables are zero. Substituting $x := x - l$, $d := d - A \cdot l$ and $r := r - l$ yields the equivalent problem

$$A \cdot x = d \quad \text{and} \quad 0 \leq x \leq r.$$

Here, x is a vector in $\mathbb{Z}^n$ such that $0 \leq x_i \leq r_i$. This shows that we may assume that the lower bound l is zero.

Furthermore, we assume that $r_i > 0$ for $i \in n$. Otherwise, if there exists an $i \in n$ such that $r_i = 0$ or $r_i < 0$, it follows that $x_i = 0$ or $x_i < 0$, respectively. In the first case the variable x_i can be removed from the system of Diophantine linear equations, whereas in the second case we see immediately that the system has no solution.

For the above system 7.2.6 with lower bound 0 and arbitrary upper bounds $r \in \mathbb{Z}^n$ on the variables we introduce a modified version of the lattice 7.2.4. The basis of the new lattice consists of the columns of the following $(m + n + 1) \times (n + 1)$-matrix:

$$\left(\begin{array}{c|cccc} -N \cdot d & & N \cdot A & & \\ \hline -r_{\max} & 2c_0 & 0 & \cdots & 0 \\ -r_{\max} & 0 & 2c_1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -r_{\max} & 0 & \cdots & \cdots & 2c_{n-1} \\ \hline r_{\max} & 0 & \cdots & \cdots & 0 \end{array}\right). \tag{7.2.7}$$

The entries $r_{\max}$ and c_i are defined by

$$r_{\max} = \operatorname{lcm}\{r_0, \ldots, r_{n-1}\} \quad \text{and} \quad c_i = \frac{r_{\max}}{r_i}, \quad i \in n,$$

and, as usual, $N \in \mathbb{N}$ is a large integer constant. In 7.6.17, we will compute a lower bound on the size of N.

After applying lattice basis reduction (see Section 7.6), the first $n - m + 1$ vectors of a reduced basis will have only zeros in the first m entries, provided N is large enough. These are relatively short vectors. The remaining m vectors contain at least one nonzero entry in the first m entries. Since entries in the first m rows are multiples of the large integer constant N, these vectors are long vectors. Thus, the new generator matrix of the lattice spanned by the columns of 7.2.7 has the following form:

$$\begin{array}{r}m\text{ rows}\left\{\vphantom{\begin{array}{c}0\\ \vdots\\ 0\end{array}}\right.\\ n+1\text{ rows}\left\{\vphantom{\begin{array}{c}\\ *\\ \end{array}}\right.\\ \\ \end{array}
\begin{array}{c}\left(\begin{array}{ccc|ccc}0&\cdots&0&*&\cdots&*\\ \vdots&&\vdots&\vdots&&\vdots\\ 0&\cdots&0&*&\cdots&*\\ \hline &&&&&\\ &*&&&*&\\ &&&&&\end{array}\right)\\ \underbrace{\hphantom{0\quad\cdots\quad 0}}_{n-m+1\text{ columns}}\;\underbrace{\hphantom{*\quad\cdots\quad *}}_{m\text{ columns}}\end{array}
\begin{array}{l}\left.\vphantom{\begin{array}{c}0\\ \vdots\\ 0\end{array}}\right\}\begin{array}{l}\text{entries are}\\ \text{multiples of }N\end{array}\\ \\ \\ \end{array}\tag{7.2.8}$$

The last m vectors cannot contribute to a solution of our original problem. Hence they can be removed from the basis. From the remaining $n-m+1$ vectors we can delete the first m entries which are zero. This gives a basis $b^{(0)}$, $b^{(1)}, \ldots, b^{(n-m)} \in \mathbb{Z}^{n+1}$ of the kernel of 7.2.6.

Theorem *With the above definitions, let* **7.2.9**

$$v = u_0 \cdot b^{(0)} + u_1 \cdot b^{(1)} + \ldots + u_{n-m} \cdot b^{(n-m)} \tag{7.2.10}$$

be an integer linear combination of the basis vectors with $v_n = r_{\max}$. The vector v is a solution of the system $A \cdot x = d$, $0 \le x \le r$, if and only if

$$v \in \mathbb{Z}^{n+1} \quad \textit{where} \quad -r_{\max} \le v_i \le r_{\max},\ i \in n\ . \tag{7.2.11}$$

Proof: Let $v = u_0 \cdot b^{(0)} + u_1 \cdot b^{(1)} + \ldots + u_{n-m} \cdot b^{(n-m)}$ be an integer linear combination of the basis vectors with $v_n = r_{\max}$. By looking at the initial basis 7.2.7 of the lattice we see that for every $i \in n$ there is an integer y_i such that $v_i = -r_{\max} + 2y_i c_i$.

By using the definitions of $r_{\max}$ and c_i it is easy to verify that $-r_{\max} \le v_i \le r_{\max}$ is equivalent to $0 \le y_i \le r_i$, $i \in n$. □

In a second step, the algorithm will search in the lattice of integer linear combinations of the basis vectors $b^{(0)}, b^{(1)}, \ldots, b^{(n-m)} \in \mathbb{Z}^{n+1}$. In this step, all lattice vectors which correspond to solutions of the original problem $A \cdot x = d$ are enumerated. Only solutions to 7.2.6 with $x_0 = 1$ are enumerated.

7.3

7.3 Basic Theory of Lattices

Let $L \subset \mathbb{R}^n$ be a lattice with basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$. We want to estimate how short the vectors of a lattice basis for L can be. For this purpose, we introduce the determinant of a lattice. We will see that the determinant has a geometrical interpretation.

7.3.1 **Definition** If a set $S \subset \mathbb{R}^n$ is measurable in the sense of Lebesgue, then its Lebesgue measure is called the volume of S and denoted by Vol_S. ◇

7.3.2 **Definition (fundamental parallelotope)** Let L be a lattice with basis $b^{(0)}$, $b^{(1)}$, $\ldots, b^{(m-1)}$. The set

$$F_B := \left\{ \sum_{i \in m} x_i b^{(i)} \mid 0 \leq x_i < 1, i \in m \right\}$$

is the *fundamental parallelotope* of the lattice L with respect to the basis $b^{(0)}$, $b^{(1)}$, $\ldots, b^{(m-1)}$. ◇

If $m = n$, i.e. if the lattice has full rank, the volume of the fundamental parallelotope F_B is equal to the absolute value of the determinant of the matrix $B = (b^{(0)} \mid \ldots \mid b^{(n-1)})$. If $m < n$, i.e. if the lattice is embedded in a space of higher dimension, the volume of the fundamental parallelotope in $\mathbb{R}^n$ is 0. Nevertheless, we will need the volume of the lattice L as a subset of the m-dimensional space $\langle b^{(0)}, b^{(1)}, \ldots, b^{(m-1)} \rangle \subset \mathbb{R}^n$. For this we introduce the *Gram matrix* $G(B)$ of the basis B.

7.3.3 **Definition (Gram matrix, determinant)** Let $B = (b^{(0)} \mid \ldots \mid b^{(m-1)})$ be a generator matrix of a lattice L with basis $b^{(0)}, \ldots, b^{(m-1)}$.

- The matrix
$$G(B) = \left(\langle b^{(i)}, b^{(j)} \rangle \right)_{i,j \in m} \in \mathbb{R}^{m \times m}$$
is called *Gram-matrix* $G(B)$ of the lattice basis.
- The *determinant* of the lattice L with respect to the generator matrix B is
$$\det(L) = \sqrt{\det(G(B))}\,.$$
◇

It is easy to see that $\det(L)$ is well-defined and that it is equal to the volume of the fundamental parallelotope F_B in the space $\langle b^{(0)}, b^{(1)}, \ldots, b^{(m-1)} \rangle$.

It is well-known that the volume of the fundamental parallelotope of a lattice does not depend on the choice of the basis (cf. Fig. 7.2). Let $m \in \mathbb{Z}$, $m > 0$. A matrix $M \in \mathbb{Z}^{m \times m}$ with determinant ± 1 is called *unimodular*.

Lemma *The volume of the fundamental parallelotope of a lattice $L \subset \mathbb{R}^n$ of rank m is equal for all bases $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ of L.* **7.3.4**

Proof: Let $A = \big(a^{(0)} \mid a^{(1)} \mid \ldots \mid a^{(m-1)}\big)$ and $B = \big(b^{(0)} \mid b^{(1)} \mid \ldots \mid b^{(m-1)}\big)$ be two generator matrices of the lattice L with fundamental parallelotopes F_A and F_B, respectively. Thus, we can express each basis vector in $\{b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}\}$ as an integer linear combination of basis vectors in $\{a^{(0)}, a^{(1)}, \ldots, a^{(m-1)}\}$, and vice versa. That is, there exists a matrix $M \in \mathbb{R}^{m \times m}$ which describes the change from generator matrix A to generator matrix B with $B = A \cdot M$. The change from generator matrix B to A can then be expressed by $A = B \cdot M^{-1}$.

Since every lattice vector is an integer linear combination of basis vectors, the entries of the matrix M as well as the entries of the matrix M^{-1} are integers. Thus, also $\det(M)$ and $\det(M)^{-1} = \det(M^{-1})$ are integers. Since $\det(M) \neq 0$, the only possibility is that $\det(M) = \pm 1$. For the volume of the fundamental parallelotopes this gives

$$\mathrm{Vol}_{F_B} = \sqrt{\det(G(B))} = \sqrt{\det(M)^2 \cdot \det(G(A))} = \mathrm{Vol}_{F_A} . \qquad \square$$

Let $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ be a basis of a lattice $L \subset \mathbb{R}^n$ of rank m. From the above proof we see that the columns of the matrix $M \cdot \big(b^{(0)} \mid b^{(1)} \mid \ldots \mid b^{(m-1)}\big)$ form another basis of L provided that $M \in \mathbb{Z}^{m \times m}$ is a unimodular matrix. This means that there is a one-to-one correspondence between the unimodular matrices and the different bases of L.

A different kind of invariant of a lattice are the *successive minima* of Minkowski [150]. Again, this invariant does not depend on the choice of the basis.

Definition (successive minima) Let $L \subset \mathbb{R}^n$ be a lattice of rank m. For an integer $i \in m$ let $\lambda_i(L)$ be the least positive real number for which there exist $i+1$ linearly independent lattice vectors $v \in L \setminus \{0\}$ with $\|v\|_2 \leq \lambda_i(L)$. The numbers $\lambda_0(L), \lambda_1(L), \ldots, \lambda_{m-1}(L)$ are the *successive minima* of the lattice L. ⋄ **7.3.5**

From the definition it follows that

$$\lambda_0(L) \leq \lambda_1(L) \leq \ldots \leq \lambda_{m-1}(L) .$$

Linearly independent vectors $v^{(i)} \in L$ with $\|v^{(i)}\| = \lambda_i(L)$ for $i \in m$ do not necessarily form a basis of the lattice. For example, the lattice

$$L = \left\{ u_0 e^{(0)} + u_1 e^{(1)} + \ldots + u_{n-1} e^{(n-1)} + u_n (\tfrac{1}{2}, \ldots, \tfrac{1}{2})^\top \mid u_0, u_1, \ldots, u_n \in \mathbb{Z} \right\}$$

in $\mathbb{Q}^n$ contains the vectors $e^{(0)}, e^{(1)}, \ldots, e^{(n-1)}$. Therefore, the successive minima of L are

$$\lambda_0(L) = \lambda_1(L) = \ldots = \lambda_{n-1}(L) = 1\,.$$

These successive minima are unique since the vectors $e^{(i)}$, $i \in n$, are the only vectors in L with Euclidean norm equal to one. But the vectors $e^{(0)}, e^{(1)}, \ldots, e^{(n-1)}$ do not form a basis of L.

The connection to quadratic forms. The arithmetic theory of lattices is closely related to the theory of positive definite quadratic forms whose long history dates back to Lagrange [120], Legendre [122], Gauss [66], Hermite [86] and Korkine and Zolotarev [112], [113].

Gauss [65] was first to notice the close connection between positive definite quadratic forms and lattices, i.e. the viewpoint of geometry. This geometric point of view was later developed systematically by Minkowski [150] and is now known as the "geometry of numbers".

7.3.6 **Definition (positive definite quadratic form)** A *positive definite quadratic form* is a map

$$f_A : \mathbb{Z}^m \to \mathbb{R} : \; x \mapsto x^\top \cdot A \cdot x\,,$$

where $A \in \mathbb{R}^{m \times m}$ is a symmetric positive definite matrix, i.e. $A^\top = A$ and $x^\top \cdot A \cdot x > 0$ for $x \in \mathbb{R}^n \setminus \{0\}$. ◇

Let $B \in \mathbb{R}^{n \times m}$ be a matrix of rank m with $m \leq n$. Setting $A := B^\top \cdot B$, we note that $f_A(x) = x^\top \cdot (B^\top \cdot B) \cdot x = \|B \cdot x\|_2^2 \geq 0$ for $x \in \mathbb{Z}^m$. Since A has maximal rank m, $f_A(x) = 0$ is equivalent to $x = 0$. It follows that the matrix A is symmetric and positive definite. Therefore, the minimum value of $f_A(x)$ for all $x \in \mathbb{Z}^m \setminus \{0\}$ is equal to the square of the ℓ_2-shortest vector in the lattice with generator matrix B.

It is well-known that for any symmetric positive definite matrix $A \in \mathbb{R}^{m \times m}$ there exists a matrix $B \in \mathbb{R}^{m \times m}$ such that $A = B^\top \cdot B$. This is known as the Cholesky decomposition of A (see [39], for instance). This shows that for every positive definite quadratic form f_A there exists a lattice L, namely the lattice whose generator matrix is the matrix B with $A = B^\top \cdot B$.

Indeed many results in lattice theory were first formulated in the language of positive definite quadratic forms. An example is 7.5.4.

Exercises

E.7.3.1 **Exercise** Prove that the volume of a fundamental parallelotope of a lattice L is equal to $\det(L)$.

Exercise Let A be a symmetric, positive definite matrix $\in \mathbb{R}^{m \times m}$. Show that there exists a matrix $B \in \mathbb{R}^{m \times m}$ with $A = B^\top \cdot B$. **E.7.3.2**

7.4 Gram–Schmidt Orthogonalization

7.4

Definition (orthogonal vectors) A set of vectors $v^{(0)}, \ldots, v^{(m-1)} \in \mathbb{R}^n \setminus \{0\}$ is called *orthogonal* if for $i, j \in m$ **7.4.1**

$$\langle v^{(i)}, v^{(j)} \rangle \begin{cases} \neq 0, & \text{if } i = j, \\ = 0, & \text{if } i \neq j\,. \end{cases}$$

◇

Lemma (Gram–Schmidt orthogonalization) *Let $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ be a set of linearly independent vectors $\in \mathbb{R}^n$. For $i = 0, 1, \ldots, m-1$, define vectors* **7.4.2**

$$\hat{b}^{(i)} = b^{(i)} - \sum_{j=0}^{i-1} \mu_{ij} \cdot \hat{b}^{(j)}\,,$$

where

$$\mu_{ij} = \frac{\langle b^{(i)}, \hat{b}^{(j)} \rangle}{\langle \hat{b}^{(j)}, \hat{b}^{(j)} \rangle}\,.$$

Then $\hat{b}^{(0)}, \hat{b}^{(1)}, \ldots, \hat{b}^{(m-1)}$ are orthogonal.

Proof: Let $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ be a set of linearly independent vectors $\in \mathbb{R}^n$. Then, $\hat{b}^{(0)} = b^{(0)}$ and $\hat{b}^{(1)} = b^{(1)} - \frac{\langle b^{(1)}, \hat{b}^{(0)} \rangle}{\langle \hat{b}^{(0)}, \hat{b}^{(0)} \rangle} \cdot \hat{b}^{(0)}$. Therefore,

$$\langle \hat{b}^{(1)}, \hat{b}^{(0)} \rangle = \langle b^{(1)}, \hat{b}^{(0)} \rangle - \frac{\langle b^{(1)}, \hat{b}^{(0)} \rangle}{\langle \hat{b}^{(0)}, \hat{b}^{(0)} \rangle} \cdot \langle \hat{b}^{(0)}, \hat{b}^{(0)} \rangle = 0\,.$$

By induction, it follows for $2 \leq k \leq m-1$ that

$$\begin{aligned} \langle \hat{b}^{(k)}, \hat{b}^{(j)} \rangle &= \langle b^{(k)}, \hat{b}^{(j)} \rangle - \sum_{i=0}^{k-1} \frac{\langle b^{(k)}, \hat{b}^{(i)} \rangle}{\langle \hat{b}^{(i)}, \hat{b}^{(i)} \rangle} \cdot \langle \hat{b}^{(i)}, \hat{b}^{(j)} \rangle \\ &= \langle b^{(k)}, \hat{b}^{(j)} \rangle - \frac{\langle b^{(k)}, \hat{b}^{(j)} \rangle}{\langle \hat{b}^{(j)}, \hat{b}^{(j)} \rangle} \cdot \langle \hat{b}^{(j)}, \hat{b}^{(j)} \rangle \\ &= 0\,, \end{aligned}$$

for $j = 0, 1, \ldots, k-1$. □

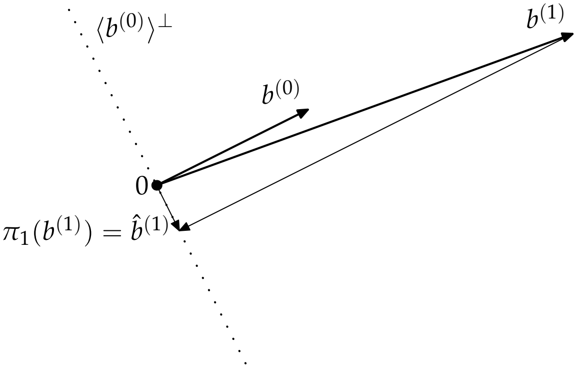

$\langle b^{(0)}\rangle^\perp$
$b^{(1)}$
$b^{(0)}$
0
$\pi_1(b^{(1)}) = \hat{b}^{(1)}$

Fig. 7.4 Orthogonal projection π_1 of $b^{(1)}$ into $\langle b^{(0)}\rangle^\perp$

The procedure 7.4.2 is called *Gram–Schmidt orthogonalization*. The vectors $\hat{b}^{(i)}$, $i \in m$, are referred to as *Gram–Schmidt vectors* and the numbers μ_{ij}, $0 \leq j \leq i < m$, are called *Gram–Schmidt coefficients*. We note that in general the set of orthogonal vectors $\hat{b}^{(0)}, \hat{b}^{(1)}, \ldots, \hat{b}^{(m-1)}$ is not longer contained in L, since the Gram–Schmidt coefficients μ_{ij} are not necessarily integers.

For $i \in m$ we can think of $\hat{b}^{(i)}$ as the orthogonal projection of $b^{(i)}$ into the subspace $H_{i-1} := \langle b^{(0)}, b^{(1)}, \ldots, b^{(i-1)}\rangle^\perp$, which is the subspace of dimension $m-i$ orthogonal to $\langle b^{(0)}, b^{(1)}, \ldots, b^{(i-1)}\rangle$ in $\langle b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}\rangle$.

7.4.3 **Definition (orthogonal projection)** With the above notation, for $t \in m$ the *orthogonal projection* $\pi_t(v)$ is defined by

$$\pi_t : \mathbb{R}^n \longrightarrow \langle b^{(0)}, b^{(1)}, \ldots, b^{(t-1)}\rangle^\perp, \quad v \mapsto \sum_{j=t}^{m-1} \frac{\langle v, \hat{b}^{(j)}\rangle}{\langle \hat{b}^{(j)}, \hat{b}^{(j)}\rangle} \cdot \hat{b}^{(j)}\,. \qquad \diamond$$

We note that the orthogonal projection depends on the choice of the basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ of the lattice $L \subset \mathbb{R}^n$. Further, from the definition it can be seen that for $t \in m$ the orthogonal projection $\pi_t(v)$ of a vector $v \in \mathbb{R}^n$ is a linear combination of the Gram–Schmidt vectors $\hat{b}^{(t)}, \ldots, \hat{b}^{(m-1)}$. For any lattice basis and any vector $v \in \mathbb{R}^n$ we have $\pi_0(v) = v$.

The orthogonal projection π_t is a linear mapping. Therefore, the projection of the lattice $L(b^{(0)}, b^{(1)}, \ldots, b^{(m-1)})$ into $\langle b^{(0)}, b^{(1)}, \ldots, b^{(t-1)}\rangle^\perp$ is again a lattice

$$\begin{aligned} L_t\big(\pi_t(b^{(t)}), \ldots, \pi_t(b^{(m-1)})\big) &:= \left\{\sum_{i\in m} u_i \pi_t(b^{(i)}) \mid u_i \in \mathbb{Z}\right\} \\ &= \left\{\sum_{i=t}^{m-1} u_i \pi_t(b^{(i)}) \mid u_i \in \mathbb{Z}\right\} \end{aligned}$$

spanned by the basis $\pi_t(b^{(t)}), \pi_t(b^{(t+1)}), \ldots, \pi_t(b^{(m-1)})$ for $t \in m$. The rank of the lattice L_t is equal to $m - t$.

In matrix notation, the Gram–Schmidt orthogonalization can be written as

$$B = \hat{B} \cdot \mu^\top$$

with a lower triangular $m \times m$-matrix

$$\mu = \begin{pmatrix} \mu_{00} & & & \\ \mu_{10} & \mu_{11} & & \\ \vdots & \vdots & \ddots & \\ \mu_{m-1,0} & \mu_{m-1,1} & \cdots & \mu_{m-1,m-1} \end{pmatrix},$$

where $\mu_{ii} = 1$, $i \in m$, and $\mu_{ij} = 0$ for $0 \leq i < j < m$. This shows that the Gram–Schmidt orthogonalization is a unimodular transformation. In particular, $\det(\mu) = 1$ and we see that we can compute the determinant $\det(L)$ from the Gram–Schmidt vectors $\hat{b}^{(0)}$, $\hat{b}^{(1)}, \ldots, \hat{b}^{(m-1)}$ of a lattice basis $b^{(0)}$, $b^{(1)}, \ldots,$ $b^{(m-1)}$ via

$$\det(L) = |\det(\hat{B}) \cdot \det(\mu)| = \prod_{i \in m} \|\hat{b}^{(i)}\|_2 \,. \qquad \mathbf{7.4.4}$$

We note that the orthogonal basis $\hat{b}^{(0)}, \hat{b}^{(1)}, \ldots, \hat{b}^{(m-1)}$ depends on the ordering of the basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$.

Exercises

Exercise Prove that the orthogonal projection is a linear mapping. **E.7.4.1**

Exercise Show that for vectors $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)} \in \mathbb{R}^n$: **E.7.4.2**

$$\sqrt{\det G(b^{(0)}, b^{(1)}, \ldots, b^{(n-1)})} = |\det(b^{(0)}, b^{(1)}, \ldots, b^{(n-1)})| \,.$$

Exercise Use elementary row and column transformations to bring the Gram matrix $G(b^{(0)}, b^{(1)}, \ldots, b^{(m-1)})$, of linearly independent $b^{(0)}$, $b^{(2)}, \ldots, b^{(m-1)} \in \mathbb{R}^n$, with $m \leq n$, to the form $\left(\langle \hat{b}^{(j)}, \hat{b}^{(l)} \rangle\right)_{j,l \in m}$. **E.7.4.3**

7.5 Bounds on Lattice Vectors

7.5

The Hadamard inequality is a well-known lower bound on the length of the vectors of a lattice basis.

7.5.1 **Lemma (Hadamard's Inequality)** *If $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$ is a basis of a lattice $L \subset \mathbb{R}^n$, then*

$$\det(L) \le \prod_{i \in m} \|b^{(i)}\|_2 .$$

Proof: Using 7.4.2 together with the mutual orthogonality of the vectors $\hat{b}^{(j)}$ we have

$$\|b^{(i)}\|_2^2 = \|\hat{b}^{(i)}\|_2^2 + \sum_{j=0}^{i-1} \mu_{ij}^2 \|\hat{b}^{(j)}\|_2^2 \ge \|\hat{b}^{(i)}\|_2^2 .$$

With 7.4.4, $\det(L) = \prod_{i \in m} \|\hat{b}^{(i)}\|_2$, the inequality follows. □

7.5.2 **Remark** The inequality of Hadamard can be written as

$$1 \le \prod_{i \in m} \frac{\|b^{(i)}\|_2}{\|\hat{b}^{(i)}\|_2} = \frac{1}{\det(L)} \cdot \prod_{i \in m} \|b^{(i)}\|_2$$

with equality if and only if the basis $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$ is orthogonal. Therefore, the product $\prod_{i \in m} \|b^{(i)}\|_2 / \|\hat{b}^{(i)}\|_2$ is a measure of the "non-orthogonality" of a basis $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$.

The inequality of Hadamard is trivially satisfied if the vectors $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$ are linearly dependent. ◇

7.5.3 **Definition (orthogonality defect)** For a lattice basis $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$

$$\prod_{i \in m} \frac{\|b^{(i)}\|_2}{\|\hat{b}^{(i)}\|_2}$$

is called *orthogonality defect* of $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$. ◇

Since $\det(L) = \prod_{i \in m} \|\hat{b}^{(i)}\|_2$ does not depend on the choice of the lattice basis, the orthogonality defect is a measure for the geometric mean of the Euclidean length of the basis vectors. Consequently, a basis consisting of short vectors has a small orthogonality defect.

A classical result due to Hermite [86] gives an upper bound for the ℓ_2-shortest vector of a lattice.

7.5.4 **Theorem (Hermite)** *Let $L \subset \mathbb{Z}^n$ be a lattice of rank m. Then L contains a nonzero vector v such that*

$$\|v\|^2 \le (4/3)^{(m-1)/2} \cdot \det(L)^{2/m} ,$$

where $\| - \|$ denotes the Euclidean norm.

Proof: Using the first successive minimum, the claim of the theorem becomes

$$\lambda_0(L)^2 \leq (4/3)^{(m-1)/2} \cdot \det(L)^{2/m}.$$

Let B be a generating matrix of the lattice L and $y \in \mathbb{Z}^m \setminus \{0\}$ be a vector for which $B \cdot y$ takes on its minimum value, i.e. $\|B \cdot y\| = \lambda_0(L)$. Then we know that $r = \gcd(y_0, y_1, \ldots, y_{m-1}) = 1$. For otherwise, since $\frac{1}{r} \cdot y$ is integral,

$$\|B \cdot (\frac{1}{r}y)\| = \frac{1}{r}\|B \cdot y\| = \frac{1}{r} \cdot \lambda_0(L),$$

which would contradict the minimality of $\lambda_0(L)$ in the case $r > 1$.

By induction it is easy to show that there exists a matrix $W \in \mathbb{Z}^{m \times m}$ such that $\det(W) = 1$ and the first column of W is equal to y (see Exercise 7.5.1). Such a matrix is unimodular and hence by the remark after 7.3.4, $B \cdot W$ is another generator matrix of L. Moreover,

$$\|B \cdot W \cdot (1,0,\ldots,0)^\top\| = \|B \cdot y\| = \lambda_0(L).$$

Therefore, we can assume that B is a generator matrix of the lattice L and that the successive minimum $\lambda_0(L)$ is attained for the first column of B, i.e. the first basis vector. Now, we search for a matrix $S \in \mathbb{R}^{m \times m}$ with $S^\top \cdot S = I_m$ and $a_0, a_1, \ldots, a_{m-1} \in \mathbb{R}$ such that

$$S \cdot B = \left(\begin{array}{c|ccc} a_0 & a_1 & \ldots & a_{m-1} \\ \hline 0 & & & \\ \vdots & & B' & \\ 0 & & & \end{array}\right).$$

The matrix S can be constructed by taking as first row the entries of the first column of B divided by $\|b^{(0)}\|$. The remaining rows are filled with linearly independent vectors in $\mathbb{R}^m$. After that, the rows of B must be orthogonalized by the Gram–Schmidt process and scaled to have norm one. Then, for arbitrary $x \in \mathbb{R}^m$ we have that

$$\begin{aligned} \|B \cdot x\|^2 &= \langle Bx, Bx \rangle \\ &= x^\top B^\top S^\top S B x \\ &= \langle SBx, SBx \rangle \\ &= \left\| \left(\begin{array}{c|ccc} a_0 & a_1 & \ldots & a_{m-1} \\ \hline 0 & & & \\ \vdots & & B' & \\ 0 & & & \end{array}\right) \cdot x \right\|^2 \\ &= (\sum_{i \in m} a_i x_i)^2 + \|B' \cdot (x_1, \ldots, x_{m-1})^\top\|^2. \end{aligned}$$

Since $a_0 = \frac{1}{\|b^{(0)}\|} b^{(0)^\top} \cdot b^{(0)} = \|b^{(0)}\|$ and $\|b^{(0)}\|$ is minimal, it follows that $a_0 = \lambda_0(L)$. For the lattice L' which is generated by B', the determinant is

$$\det(L)^2 = \det(G(B)) = \det(G(SB)) = a_0^2 \cdot \det(G(B')) = a_0^2 \cdot \det(L')^2 .$$

Therefore,

$$\det(L') = \det(L) \cdot \frac{1}{\lambda_0(L)} .$$

We can now prove the initial claim by induction. The case $m = 1$ is clear. For $m > 1$, suppose that there exists a vector $x' = (x_1, \ldots, x_{m-1})^\top \in \mathbb{Z}^{m-1}$ with

$$\begin{aligned} \|B' \cdot x'\|^2 &\leq (4/3)^{(m-2)/2} \cdot \det(L')^{2/(m-1)} \\ &= (4/3)^{(m-2)/2} \cdot \det(L)^{2/(m-1)} \frac{1}{\lambda_0(L)^{2/(m-1)}} . \end{aligned}$$

For the fixed values $x_1, \ldots, x_{m-1} \in \mathbb{Z}$ we can choose $x_0 \in \mathbb{Z}$ such that

$$|x_0 + \frac{a_1 x_1 + \ldots + a_{m-1} x_{m-1}}{a_0}| \leq \frac{1}{2} .$$

It follows that

$$\left(\sum_{i \in m} a_i x_i \right)^2 \leq \frac{1}{4} \lambda_0^2(L)$$

and therefore, since $\lambda_0^2(L) \leq \|B \cdot x\|^2$, we have

$$\lambda_0^2(L) \leq \frac{1}{4} \lambda_0^2(L) + (4/3)^{(m-2)/2} \cdot \det(L)^{2/(m-1)} \frac{1}{\lambda_0(L)^{2/(m-1)}} .$$

An easy calculation shows that

$$\lambda_0^2(L) \leq (4/3)^{(m-1)/2} \cdot \det(L)^{2/m} ,$$

which is the claim made at the beginning of the proof. Since this result is equivalent to the assertion, the proof is finished. □

Minkowski started a systematic theory which is now known as the geometry of numbers. In [149], he proved the following fundamental theorem.

7.5.5 **Theorem (Minkowski)** *Let S be a convex set in $\mathbb{R}^n$ which is symmetric about the origin (i.e. $x \in S \Rightarrow -x \in S$). If the volume of S is greater than 2^n, then S contains a nonzero vector $v \in \mathbb{Z}^n$.*

For the proof of the theorem we use the following lemma by Blichfeldt [21].

7.5.6 **Lemma (Blichfeldt)** *Let S be a measurable set in $\mathbb{R}^n$. If $Vol_S > 1$ or S is bounded and closed and $Vol_S = 1$ then S contains two different vectors x and y such that $x - y$ is in $\mathbb{Z}^n \setminus \{0\}$.*

Proof: First we assume that $\mathrm{Vol}_S > 1$. Without loss of generality we suppose that S is bounded. The volume of the cube $Q := \{x \in \mathbb{R}^n \mid 0 \leq x_i < 1,\ i \in n\}$ is equal to 1. Since S is bounded there are finitely many integral vectors $u^{(0)}, u^{(1)}, \ldots, u^{(k-1)} \in \mathbb{Z}^n$ such that the intersection of S and $u^{(i)} + Q$, $i \in k$, is nonempty. Here, for $v \in \mathbb{R}^n$ the set $v + Q$ is defined as $v + Q = \{x \in \mathbb{R}^n \mid \exists q \in Q : x = v + q\}$.

For $i \in k$ set $S_i := S \cap (u^{(i)} + Q)$ and $S_i' := S_i - u^{(i)}$, compare Fig. 7.5. Then, for $i \in k$ the sets S_i' are contained in Q. On the other hand, we have

$$\sum_{i \in k} \mathrm{Vol}_{S_i'} = \sum_{i \in k} \mathrm{Vol}_{S_i} = \mathrm{Vol}_S > 1\,.$$

Therefore, these sets cannot be mutually disjoint and there are two sets S_j', S_l', $j, l \in k$, and a vector $z \in \mathbb{R}^n$ such that

$$z \in S_j' \cap S_l'\,.$$

It follows that both $x := u^{(j)} + z$ and $y := u^{(l)} + z$ are contained in S and $x - y = u^{(j)} - u^{(l)}$ is an integral vector.

Next, we assume that S is bounded and closed and $\mathrm{Vol}_S = 1$. Let $\theta_r > 1$ be a sequence of numbers with $\lim_{r \to \infty} \theta_r = 1$. For each r the set $\theta_r S$ has volume strictly greater than 1. From the previous result it follows that for each r there exist vectors $x^{(r)}, y^{(r)} \in \theta_r S$ such that $x^{(r)} - y^{(r)}$ is a nonzero integral vector. Since S is bounded and closed there exist subsequences $(x^{(r_t)})_{t \in \mathbb{N}}$, $(y^{(r_t)})_{t \in \mathbb{N}}$ converging to some vectors x and y in S, respectively. The difference $x - y$ must be a nonzero integral vector, which proofs the assertion. □

Proof of Minkowski's theorem: Define $S/2 := \{x \in \mathbb{R}^n \mid 2x \in S\}$. Since the volume of S is greater than 2^n, $S/2$ has volume greater than 1. Using the lemma of Blichfeldt, we know that $S/2$ contains two different vectors x and y such that $x - y \in \mathbb{Z}^n \setminus \{0\}$. Therefore, $2x$ and $2y$ belong to S. Since S is symmetric about the origin, also $-2y$ belongs to S. The fact that S is convex implies that $\frac{1}{2}(2x) + \frac{1}{2}(-2y) = x - y \in S$. Since $x - y \in \mathbb{Z}^n$ this proves the theorem. □

This result is sharp, as the n-dimensional cube $\{x \in \mathbb{R}^n \mid \|x\|_\infty < 1\}$ has volume 2^n and does not contain an integral vector $\neq 0$.

Remark As we can see from the proof, the theorem of Minkowski is also valid if the set S is bounded and closed and has $\mathrm{Vol}_S = 2^n$. ◇ 7.5.7

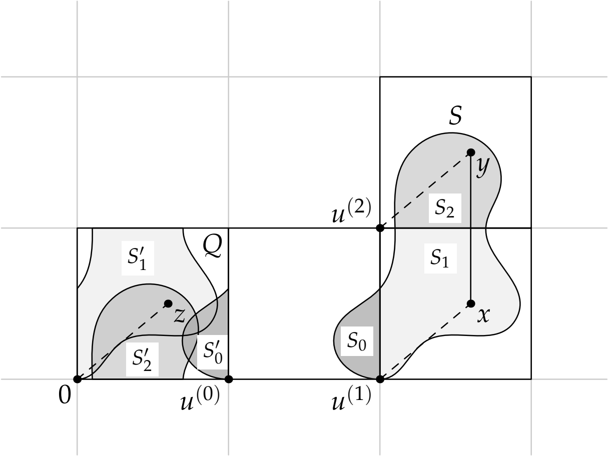


Fig. 7.5 $S_i = S \cap (u^{(i)} + Q)$ and $S_i' = S_i - u^{(i)}$, $i = 0, 1, 2$, in the Lemma of Blichfeldt

7.5.8 **Definition (Volume of the unit sphere)** We denote by ρ_n the volume of the unit sphere $S := \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$ in $\mathbb{R}^n$:

$$\rho_n = \mathrm{Vol}_S = \frac{\pi^{n/2}}{\frac{n}{2}!},$$

where $\frac{n}{2}!$ is defined by $0! = 1$, $\frac{1}{2}! = \sqrt{\pi}/2$, and $\frac{n}{2}! = \frac{n}{2} \cdot (\frac{n}{2} - 1)!$ for $n \in \mathbb{Z}$, $n > 1$. ◇

As a direct consequence of 7.5.5, we have the following bound for the Euclidean length of an ℓ_2-shortest vector in a lattice L.

7.5.9 **Theorem (Minkowski)** *If $L \subset \mathbb{R}^n$ is a lattice of rank n, then there is a nonzero vector $v \in L$ with*

$$\|v\|_2 \leq 2\left(\frac{\det(L)}{\rho_n}\right)^{1/n} = \frac{2}{\pi}\left(\frac{n}{2}! \cdot \det(L)\right)^{1/n}.$$

Proof: Let $s \in \mathbb{R}$ be a positive number and $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)}$ be a basis of the lattice L. Consider the ellipsoid $K = \{x \in \mathbb{R}^n \mid \|B \cdot x\|_2^2 \leq s\}$ which is centered at 0 and whose volume is

$$\mathrm{Vol}(K) = \frac{\rho_n \cdot s^{n/2}}{\det(L)}.$$

Choose s such that

$$\frac{\rho_n \cdot s^{n/2}}{\det(L)} = 2^n.$$

By 7.5.5, there exists a nonzero lattice vector $v \in L$ with

$$\|v\|_2^2 \le 4\left(\frac{\det(L)}{\rho_n}\right)^{2/n},$$

proving the theorem. □

7.5.9 gives an upper bound for the ratio

$$\frac{\lambda_0(L)}{\det(L)^{1/n}} \le \frac{2}{\rho_n^{1/n}}.$$

Occasionally, the weaker estimate

$$\frac{\lambda_0(L)}{\det(L)^{1/n}} \le \sqrt{n}$$

of [107] suffices.

Definition (Hermite's constant) The supremum of the ratio **7.5.10**

$$\frac{\lambda_0^2(L)}{\det(L)^{2/n}}$$

over all lattices in $\mathbb{R}^n$ of rank n is called *Hermite's constant* and is denoted as γ_n. ◇

Blichfeldt [22] provided the upper bound

$$\gamma_n \le \frac{n\big(1+o(1)\big)}{e\pi}. \tag{7.5.11}$$

Hermite's constant is known exactly for $n \le 8$. Meanwhile, the best known bounds for Hermite's constant are

$$\frac{n+\log(\pi\log n)}{2e\pi} + o(1) \le \gamma_n \le \frac{1.744n}{2e\pi}\big(1+o(1)\big).$$

The lower bound is from [148] and the upper bound is contained in [40].

Using the successive minima of a lattice, Minkowski [150] was able to sharpen the bounds of Theorems 7.5.5 and 7.5.9:

Theorem (Minkowski's Second Theorem) **7.5.12**
If $L \subset \mathbb{R}^n$ is a lattice of rank n with successive minima $\lambda_0(L), \lambda_1(L), \ldots, \lambda_{n-1}(L)$, then

$$\lambda_0(L)\lambda_1(L)\cdots\lambda_{n-1}(L) \le 2^n \det(L).$$

Proof: The proof can be found in [77, p. 59]. □

Exercises

E.7.5.1 **Exercise** Let $y_0, y_1, \dots, y_{m-1}$ be integers with

$$\gcd(y_0, y_1, \dots, y_{m-1}) = 1 .$$

Show that there exists a matrix $W \in \mathbb{Z}^{m \times m}$ with $\det(W) = 1$ whose first column is equal to $(y_0, y_1, \dots, y_{m-1})^\top$.

7.6 Lattice Basis Reduction

In this section we outline the classical concepts of lattice basis reduction as developed by Korkine and Zolotarev and later by Minkowski. Furthermore, we describe the celebrated LLL-algorithm which computes another type of reduced basis, the LLL-reduced or δ-reduced basis. We conclude this section by discussing some improvements and variations of the LLL-algorithm. Unless stated otherwise, by $\|-\|$ we always denote the Euclidean norm in this section.

Classical concepts of lattice basis reduction. Reduction methods for positive definite quadratic forms were first studied by Lagrange [120] for $n = 2$ and Gauss [66], [65] and Seeber [176] for $n = 3$. Hermite [87] was the first to propose a reduction method for positive quadratic forms for general values of n.

In his seminal work [150], Minkowski introduced the notion of a reduced basis of a lattice in dimension n for arbitrary positive integers n.

7.6.1 **Definition (Minkowski-reduced basis)** A basis $b^{(0)}, b^{(1)}, \dots, b^{(n-1)}$ of the lattice $L \subset \mathbb{R}^n$ is *reduced in the sense of Minkowski*, if for $t = 0, 1, \dots, n-1$

1. the vector $b^{(t)}$ is a shortest vector in L and
2. the set $\{b^{(0)}, b^{(1)}, \dots, b^{(t)}\}$ can be extended to a basis of L. ◇

In [151], Minkowski showed that

$$\frac{1}{\det(L)} \cdot \prod_{i \in n} \|b^{(i)}\| \leq \frac{2^n}{\rho_n} \left(\frac{3}{2}\right)^{n(n-1)/2} = 2^{O(n^2)}$$

is an upper bound for the orthogonality defect of a Minkoswki-reduced basis. This bound is much larger than the bound which can be derived from Minkowski's Second Theorem 7.5.12. If there exists a basis $b^{(0)}, b^{(1)}, \dots, b^{(n-1)}$

of the lattice L such that the lengths of the basis vectors equal the successive minima, then we can bound the orthogonality defect by

$$\frac{1}{\det(L)} \cdot \prod_{i \in n} \|b^{(i)}\| \leq 2^n \,.$$

Since the vector $b^{(0)}$ of a Minkowski-reduced basis $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)}$ of a lattice L is an ℓ_2-shortest vector in L, the computation of a Minkowski-reduced basis of a lattice L is at least as hard as computing an ℓ_2-shortest vector in L.

From a computational point of view, the reduced bases of Korkine and Zolotarev [113] have turned out to be more useful.

Definition (Korkine–Zolotarev-reduced basis) A basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ of a lattice $L \subset \mathbb{R}^n$ is *reduced in the sense of Korkine and Zolotarev* [113], if **7.6.2**

1. $b^{(0)}$ is an ℓ_2-shortest vector in L and
2. for all $t \in m$, $\hat{b}^{(t)}$ is an ℓ_2-shortest vector in the lattice $L_t(b^{(t)}, \ldots, b^{(m-1)})$. $\diamond$

The upper bound on the orthogonality defect of a Korkine–Zolotarev-reduced basis is much better than that of a Minkowski-reduced basis. Lagarias, Lenstra and Schnorr [118] proved the following bounds.

Theorem *Let $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)}$ be a Korkine–Zolotarev-reduced basis of a lattice $L \subset \mathbb{Z}^n$. Then* **7.6.3**

$$\sqrt{\frac{4}{i+4}}\lambda_i(L) \leq \|b^{(i)}\| \leq \sqrt{\frac{i+4}{4}}\lambda_i(L) \quad \textit{for } i \in n$$

and

$$\prod_{i \in n} \|b^{(i)}\| \leq \Big(\gamma_n^n \cdot \prod_{i \in n} \frac{i+4}{4}\Big)^{1/2} \cdot \det(L) \,. \qquad \square$$

Let L be a lattice in $\mathbb{Z}^n$, and let $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)}$ be a Korkine–Zolotarev-reduced basis for L. Using the asymptotic result 7.5.11 of Blichfeldt, the asymptotic upper bound for the orthogonality defect of a Korkine–Zolotarev-reduced basis $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)}$ can be shown to be of order $O(n^n)$.

The vector $b^{(0)}$ of a Korkine–Zolotarev-reduced basis $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)}$ of a lattice L is an ℓ_2-shortest vector in L. So, the computation of a Korkine–Zolotarev-reduced basis of an lattice L is at least as hard as computing an ℓ_2-shortest vector in L.

The LLL-algorithm. Summarizing, no fast algorithm to compute a Minkowski-reduced basis or a Korkine–Zolotarev-reduced basis is known. A major breakthrough was achieved by Lenstra, Lenstra, and Lovász in their seminal work

[125]. They compute a different type of reduced basis, which is now called an LLL-reduced basis. We only give a brief outline of the algorithm. For a detailed description, the reader is referred to the original paper [125] or to textbooks, like [39], for example.

Again, in this section the norm $\| - \|$ always denotes the Euclidean norm. For $r \in \mathbb{R}$, $\lfloor r \rceil$ denotes the nearest integer to r, i.e. $\lfloor r \rceil := \lfloor \frac{1}{2} + r \rfloor$.

A high-level description of the algorithm is as follows.

7.6.4 **Algorithm (LLL-algorithm** [125]**)** The LLL (or L^3) algorithm computes an LLL-reduced basis. The input is a basis $b^{(0)}, \ldots, b^{(m-1)}$ of the lattice L of rank m.

```
(1)  Let $\delta \in \mathbb{R}$ with $\frac{1}{4} < \delta < 1$.
(2)  Set $k := 0$.
(3)  do
(4)     1. for $j = 0, \ldots, k-1$
(5)           replace $b^{(k)}$ by $b^{(k)} - \lfloor \mu_{kj} \rceil b^{(j)}$,
(6)              where $\mu_{kj}$ is the Gram-Schmidt coefficient from 7.4.2.
(7)     2. if $\delta \|\pi_k(b^{(k)})\|^2 > \|\pi_k(b^{(k+1)})\|^2$ then
(8)           interchange $b^{(k+1)}$ and $b^{(k)}$
(9)           update $\hat{b}^{(k+1)}$, $\hat{b}^{(k)}$ and $\mu$
(10)          set $k := \max(k-1, 0)$
(11)       else
(12)          set $k := k+1$
(13) until $k = m-1$.
```

□

Step 1 (line (4)) of the algorithm achieves that in each stage the basis vectors are "as orthogonal as possible". This means that the Gram–Schmidt orthogonalized vector is approximated by an integer linear combination of the basis vectors, compare Fig. 7.6. The hope is that for $0 \leq i \leq k$ the basis vectors $b^{(0)}, b^{(1)}, \ldots, b^{(i-1)}$ are close to being orthogonal. That is, they are good approximations of their Gram–Schmidt vectors $\hat{b}^{(0)}, \hat{b}^{(1)}, \ldots, \hat{b}^{(i-1)}$.

In Step 2 (line (7)) of the algorithm the Euclidean length of two vectors are compared:

7.6.5
$$\delta \|\pi_k(b^{(k)})\|^2 > \|\pi_k(b^{(k+1)})\|^2 .$$

The first vector

$$\pi_k(b^{(k)}) = \hat{b}^{(k)}$$

on the left hand side of 7.6.5 is the orthogonal projection of $b^{(k)}$ onto the subspace $\langle b^{(0)}, b^{(1)}, \ldots, b^{(k-1)} \rangle^{\perp}$. The second vector

$$\pi_k(b^{(k+1)}) = \sum_{i=k}^{m-1} \mu_{k+1,i} \hat{b}^{(i)} = \hat{b}^{(k+1)} + \mu_{k+1,k} \hat{b}^{(k)}$$

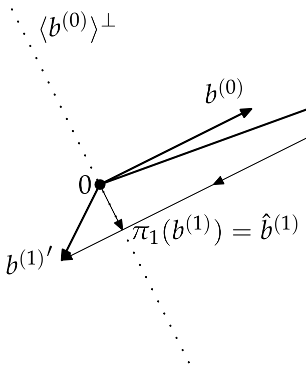
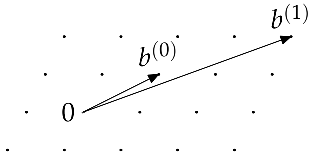

$\langle b^{(0)}\rangle^\perp$
$b^{(1)}$
$b^{(0)}$
0
$b^{(1)'}$
$\pi_1(b^{(1)}) = \hat{b}^{(1)}$

Fig. 7.6 $b^{(1)'}$ is the integer approximation of $\pi_1(b^{(1)})$

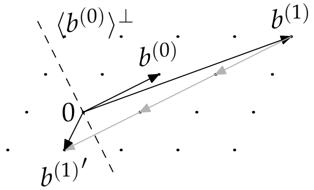
on the right hand side of 7.6.5 is the orthogonal projection of the vector $b^{(k+1)}$ into $\langle b^{(0)}, b^{(1)}, \ldots, b^{(k-1)}\rangle^\perp$. Depending on the length of their projected vectors onto $\langle b^{(0)}, b^{(1)}, \ldots, b^{(k-1)}\rangle^\perp$, we choose either $b^{(k)}$ or $b^{(k+1)}$ as the new vector $b^{(k)}$. In order to prove convergence of the algorithm, we only accept $b^{(k+1)}$ as the new basis vector $b^{(k)}$ if the length of the new orthogonal vector $\hat{b}^{(k)}$ is reduced significantly, i.e. if it is reduced by at least a factor of δ.

7.6.6 **Example** To illustrate the LLL-algorithm we consider the rank 2 lattice which is spanned by the vectors $b^{(0)} = \binom{4}{2}$ and $b^{(1)} = \binom{11}{4}$. Since $m = 2$, the variable k remains equal to zero throughout the algorithm. An LLL-reduced basis with $\delta = 1$ is computed by executing the following steps.

- 

  The input basis consisting of the vectors

$$b^{(0)} = \begin{pmatrix} 4 \\ 2 \end{pmatrix} \text{ and } b^{(1)} = \begin{pmatrix} 11 \\ 4 \end{pmatrix}.$$

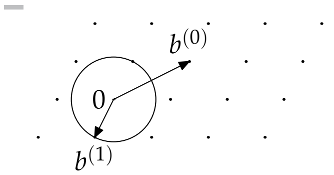
- 

  Since

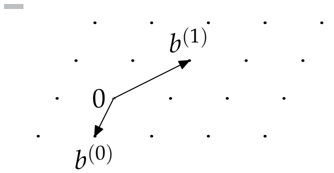
$$\mu_{10} = \frac{\langle \binom{11}{4}, \binom{4}{2}\rangle}{\langle \binom{4}{2}, \binom{4}{2}\rangle} = \frac{13}{5} = 2.6,$$

  we set $\lfloor \mu_{10} \rceil = 3$ in step 1 (line (5)). Then according to line (5), $b^{(1)}$ is replaced by

$$b^{(1)'} = \begin{pmatrix} 11 \\ 4 \end{pmatrix} - 3 \cdot \begin{pmatrix} 4 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}.$$



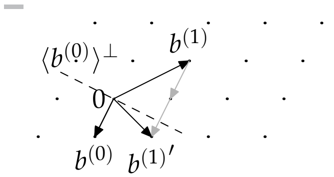
Step 2 (line (7)): $\|\pi_0(b^{(0)})\|^2 = 20$ and $\|\pi_0(b^{(1)})\|^2 = 5$ are compared.



Since

$$20 = \|\pi_0(b^{(0)})\|^2 > \|\pi_0(b^{(1)})\|^2 = 5\,,$$

the two vectors are swapped.

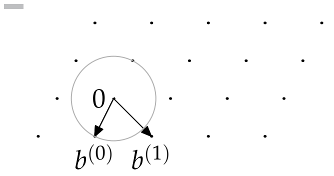


Again, in step 1 (line (5))

$$\mu_{10} = \frac{\langle\binom{4}{2},\binom{-1}{-2}\rangle}{\langle\binom{-1}{-2},\binom{-1}{-2}\rangle} = \frac{-8}{5} = -1.6.$$

Therefore, $\lfloor\mu_{10}\rceil = -2$ and the vector $b^{(1)}$ is replaced by

$$b^{(1)'} = \begin{pmatrix}4\\2\end{pmatrix} - (-2)\cdot\begin{pmatrix}-1\\-2\end{pmatrix} = \begin{pmatrix}2\\-2\end{pmatrix}.$$



Since

$$5 = \|\pi_0(b^{(0)})\|^2 < \|\pi_0(b^{(1)})\|^2 = 8\,,$$

the two vectors are not swapped in step 2 (line (7)) and the algorithm terminates. ◇

Algorithm 7.6.4 is only a very informal description of the algorithm. After swapping the two vectors $b^{(k)}$ and $b^{(k+1)}$, the Gram–Schmidt coefficients and the length of the Gram–Schmidt vectors require updating. The exact algorithm is given in [125].

The LLL-algorithm as described in Algorithm 7.6.4 works with rational numbers μ_{ij} and with rational vectors $\hat{b}^{(i)}$ for $0 \leq j \leq i < m$. Since numerator and denominator of rational numbers are integers which can be represented exactly on a computer, it was proposed in [125] that the LLL-algorithm might be formulated as an algorithm solely over the integers.

In order to represent all integers which appear in the course of the algorithm, the subdeterminants D_i of the Gram matrix can be used, i.e.

$$D_i := \det\big(\langle b^{(j)}, b^{(k)}\rangle\big)_{0\leq j,k\leq i} = \prod_{k\in i} \|\hat{b}^{(k)}\| \quad \text{for } i \in m\,.$$

We know from [125] that

- $\|\hat{b}^{(i)}\|^2 = D_i / D_{i-1}$ for $i \in m$,
- $D_{i-1}\hat{b}^{(i)} \in L \subset \mathbb{Z}^n$ for $i \in m$ (and $D_{-1} := 1$),
- $D_j \mu_{ij} \in \mathbb{Z}$ for $0 \le j < i < m$.

Then, the LLL-algorithm can be modified to work with the integers $D_j\mu_{ij}$ and the integer vectors $D_{i-1}\hat{b}^{(i)}$ instead of μ_{ij} and $\hat{b}^{(i)}$, respectively. Thus the LLL-algorithm becomes an algorithm with exact arithmetic whose time complexity can then be estimated.

For the time complexity of the original LLL-algorithm, note that each interchange of the vectors $b^{(k)}$ and $b^{(k+1)}$ for $0 \le k < m-1$ in step 2 of 7.6.4 reduces the value of $D := \prod_{i=0}^{m-2} D_i$ by at least a factor of δ. In step 1 of the algorithm, the Gram–Schmidt vectors and therefore also D remain unchanged. But, there exists a lower bound for D which is independent of the choice of the basis. This can be seen for example from 7.5.12: $D_i \ge \lambda_0(L)\lambda_1(L)\cdots\lambda_i(L)/2^i$ for $i \in m$. Therefore, Algorithm 7.6.4 terminates after a finite number of steps.

In [125] the following bounds on the time complexity of the LLL-algorithm are given.

Theorem *Let $M \in \mathbb{R}$, $M \ge 2$, be such that $\|b^{(i)}\|^2 \le M$ for $i \in m$. The number of arithmetic operations needed by the LLL-algorithm is $O(m^4 \log M)$ and the integers on which these operations are performed each have binary length $O(m \log M)$.* □ **7.6.7**

Definition (δ-reduced basis) The output $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ of the LLL-algorithm with $\frac{1}{4} < \delta < 1$ is called δ-reduced basis of the lattice L. Sometimes, it is simply called an LLL-reduced basis of the lattice L. ◇ **7.6.8**

In [125], the following bounds on the quality of the reduction are shown:

Theorem *Let $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ be a δ-reduced basis of the lattice $L \subset \mathbb{Q}^n$. Then* **7.6.9**

$$\|b^{(j)}\|^2 \le \left(\frac{4}{4\delta-1}\right)^i \cdot \|\hat{b}^{(i)}\|^2 \text{ for } 0 \le j \le i < m. \tag{7.6.10}$$

$$\det(L) \le \prod_{i\in m} \|b^{(i)}\| \le \left(\frac{4}{4\delta-1}\right)^{m(m-1)/4} \cdot \det(L). \tag{7.6.11}$$

$$\|b^{(0)}\| \le \left(\frac{4}{4\delta-1}\right)^{(m-1)/4} \cdot \det(L)^{1/m}. \tag{7.6.12}$$

Proof: Since the basis $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$ is δ-reduced, the condition of step 2 in the LLL-algorithm is not satisfied for all $0 \le k < m-1$. This together with the fact that the Gram–Schmidt vectors are mutually orthogonal implies that for $1 \le i < m$

$$\|\hat{b}^{(i)}\|^2 + \mu_{i,i-1}^2 \|\hat{b}^{(i-1)}\|^2 \ge \delta \|\hat{b}^{(i-1)}\|^2.$$

With step 1 of the LLL-algorithm we can argue that $\mu_{i,i-1}^2 \le \frac{1}{4}$, see [125]. This gives

$$\|\hat{b}^{(i)}\|^2 \ge \frac{4\delta - 1}{4} \cdot \|\hat{b}^{(i-1)}\|^2 \text{ for } 0 < i < m.$$

It follows for $0 \le j \le i < m$ that

$$\|\hat{b}^{(j)}\|^2 \le \Big(\frac{4}{4\delta - 1}\Big)^{i-j} \|\hat{b}^{(i)}\|^2 .$$

With

$$\|b^{(i)}\|^2 = \|\hat{b}^{(i)}\|^2 + \sum_{j \in i} \mu_{ij}^2 \|\hat{b}^{(j)}\|^2$$

and some elementary calculations (see Exercise 7.6.1) we get for $i \in m$:

$$\|b^{(i)}\|^2 \le \|\hat{b}^{(i)}\|^2 + \sum_{j \in i} \frac{1}{4} \Big(\frac{4}{4\delta - 1}\Big)^{i-j} \|\hat{b}^{(i)}\|^2 \le \Big(\frac{4}{4\delta - 1}\Big)^{i} \|\hat{b}^{(i)}\|^2 .$$

Hence,

$$\|b^{(j)}\|^2 \le \Big(\frac{4}{4\delta - 1}\Big)^{j} \cdot \|\hat{b}^{(j)}\|^2 \le \Big(\frac{4}{4\delta - 1}\Big)^{i} \cdot \|\hat{b}^{(i)}\|^2 .$$

Applying Hadamard's inequality 7.5.1 and 7.6.10 with $j = i$ we obtain

$$\det(L) \le \prod_{i \in m} \|b^{(i)}\| \le \prod_{i \in m} \Big(\frac{4}{4\delta - 1}\Big)^{i/2} \cdot \|\hat{b}^{(i)}\| = \Big(\frac{4}{4\delta - 1}\Big)^{\frac{m(m-1)}{4}} \cdot \det(L) ,$$

which is 7.6.11.

If we set $j := 0$ in 7.6.10 then the product of the right hand side of 7.6.10 for $i \in m$ gives $\|b^{(0)}\| \le \big(\frac{4}{4\delta-1}\big)^{(m-1)/4} \cdot \det(L)^{1/m}$. □

It follows from 7.6.11 that the orthogonality defect of a δ-reduced basis can be bounded above by

$$\frac{1}{\det(L)} \cdot \prod_{i \in m} \|b^{(i)}\| \le \Big(\frac{4}{4\delta - 1}\Big)^{m(m-1)/4} ,$$

which is $2^{O(m^2)}$ for $\delta = 3/4$. Thus, the orthogonality defect of an LLL-reduced basis has approximately the same size as the orthogonality defect of a basis which is reduced in the sense of Minkowski.

In [125], the authors provide upper bounds of the Euclidean lengths of the reduced basis vectors, compared to Euclidean lengths of a shortest lattice vector:

Theorem *Let $L \subset \mathbb{Q}^n$ be a lattice with δ-reduced basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$. Then* **7.6.13**

$$\|b^{(0)}\|^2 \leq \left(\frac{4}{4\delta - 1}\right)^{m-1} \cdot \lambda_0(L)^2 . \qquad \textbf{7.6.14}$$

Proof: Let v be a vector in L such that $\|v\| = \lambda_0(L)$. Then we can write $v = \sum_{j \in m} r_j b^{(j)} = \sum_{j \in m} r'_j \hat{b}^{(j)}$ with $r_j \in \mathbb{Z}$ and $r'_j \in \mathbb{Q}$ for $j \in m$. If t is the largest index such that $r_t \neq 0$ then we have $r_t = r'_t$ (cf. Exercise 7.6.2). Thus we deduce the inequality

$$\lambda_0^2(L) \geq r'^2_t \|\hat{b}^{(t)}\|^2 \geq \|\hat{b}^{(t)}\|^2 .$$

With 7.6.10 we have the bound

$$\|b^{(0)}\|^2 \leq (\frac{4}{4\delta - 1})^t \cdot \|\hat{b}^{(t)}\|^2 \leq (\frac{4}{4\delta - 1})^{m-1} \cdot \|\hat{b}^{(t)}\|^2 .$$

Combining the two inequalities gives the required bound for $\|b^{(0)}\|^2$. □

At first sight, the bound in 7.6.13 on the Euclidean length of the first basis vector of a LLL-reduced lattice basis does not look promising. However, there are situations where this theoretical bound is already good enough, i.e. where any nonzero vector in L which is not an ℓ_2-shortest vector has Euclidean length greater than $(\frac{4}{4\delta-1})^{(m-1)/2} \cdot \lambda_0(L)$. Problems of this type can be solved by the LLL-algorithm in polynomial time. Examples are attacks on knapsack based cryptosystems with low-density [119], [43].

Secondly, in nearly all practical situations the LLL-algorithm behaves much better than the bound 7.6.14 indicates. It was already noted in [125] that the bound $\left(\frac{4}{4\delta-1}\right)^{m-1}$ in 7.6.13, which proved to be rather pessimistic in most instances, can be replaced by $\max\{\|b^{(i)}\|_2^2 / \|\hat{b}^{(j)}\|_2^2 \mid 0 \leq i \leq j < m\}$. If an LLL-reduced basis is available, then computing this bound is trivial. For $i = 0$ in many cases this bound turns out to be close to 1 and hence $b^{(0)}$ actually is an ℓ_2-shortest vector in the lattice.

On the other hand, Kannan [106] notes that there are lattices L of rank m for which the orthogonality defect of certain LLL-reduced bases reaches the bound 7.6.11 and the square of the norm of the first basis vector is larger than $\lambda_0(L)$ by a factor $2^{O(m^2)}$.

The following generalizations of 7.6.13 can be found in [125].

Theorem *Let $L \subset \mathbb{Q}^n$ be a lattice with LLL-reduced basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$. For $t \in m$ let $v^{(0)}, v^{(1)}, \ldots, v^{(t-1)} \in L$ be t linearly independent lattice vectors. Then we have* **7.6.15**

$$\|b^{(t)}\|^2 \leq \left(\frac{4}{4\delta - 1}\right)^{m-1} \cdot \max\{\|v^{(i)}\|^2 \mid i \in t\} .$$

Proof: See [125, Prop. 1.12]. □

We note that if a lattice basis $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$ of a lattice $L \subset \mathbb{Q}^n$ is δ-reduced with $\frac{1}{4} < \delta < 1$ then for $1 \le t \le m$ also the lattices $L_t(b^{(t)}, \dots, b^{(m-1)})$ are δ-reduced. Moreover, the vector $\hat{b}^{(t)} = \pi_t(b^{(t)})$ is the first vector of the lattice basis $\pi_t(b^{(t)}), \dots, \pi_t(b^{(m-1)})$ of the lattice $L_t(b^{(t)}, \dots, b^{(m-1)})$. Applying 7.6.12 and 7.6.14 to the lattice $L_t(b^{(t)}, \dots, b^{(m-1)})$ we get for $t \in m$:

7.6.16 **Corollary** *Let $L \subset \mathbb{Q}^n$ be a lattice with δ-reduced basis $b^{(0)}, b^{(1)}, \dots, b^{(m-1)}$. Then, for $t \in m$:*

$$\|\hat{b}^{(t)}\| \le \left(\frac{4}{4\delta - 1}\right)^{(m-1-t)/2} \cdot \lambda_0\big(L_t(b^{(t)}, \dots, b^{(m-1)})\big),$$

$$\|\hat{b}^{(t)}\| \le \left(\frac{4}{4\delta - 1}\right)^{(m-1-t)/4} \cdot \det\big(L_t(b^{(t)}, \dots, b^{(m-1)})\big)^{1/(m-t)}. \qquad \square$$

The upper bounds on the basis vectors in 7.6.15 can be used to determine the size of the integer constant N in 7.2.7.

7.6.17 **Theorem** *Let $A \cdot x = d$ with $A \in \mathbb{Z}^{m \times n}$ and $d \in \mathbb{Z}^m$. There exists a constant N which depends only on the size of the entries of A and d such that the LLL-algorithm computes a basis of the form 7.2.8 if applied to the basis 7.2.7.*

Proof: Without loss of generality, we assume that the matrix A has rank m. Thus we can permute the columns of A such that the first m columns, i.e. $A^{(0)}, A^{(1)}, \dots, A^{(m-1)}$, are linearly independent. Let $A' = \big(A^{(0)} \mid \dots \mid A^{(m-1)}\big)$. Then, each of the $n - m + 1$ linear systems

$$A' \cdot x = -A^{(m+i)}, \quad 0 \le i < n - m,$$

and

$$A' \cdot x = d,$$

possesses a unique solution in $\mathbb{Q}^m$.

For $0 \le i < n - m$ let $v'^{(i)} \in \mathbb{Q}^m$ be the solution of the system of linear equations $A' \cdot x = -A^{(m+i)}$ and $v'^{(n-m)} \in \mathbb{Q}^m$ be the solution of the system $A' \cdot x = d$. Using Cramer's rule we can explicitly compute the solutions $v'^{(i)}$ for $0 \le i < n - m$ via

$$v_k'^{(i)} = \frac{1}{\det(A')} \cdot \det\big((A^{(0)}, \dots, A^{(k-1)}, -A^{(m+i)}, A^{(k+1)}, \dots, A^{(m-1)})\big),\ k \in m,$$

and

$$v_k'^{(n-m)} = \frac{1}{\det(A')} \cdot \det\big((A^{(0)}, \dots, A^{(k-1)}, d, A^{(k+1)}, \dots, A^{(m-1)})\big),\ k \in m.$$

Setting $M := \max\{\|A^{(0)}\|, \|A^{(1)}\|, \dots, \|A^{(n-1)}\|, \|d\|\}$ and using Hadamard's inequality 7.5.1, we can bound the entries $|v_k'^{(i)}|$, $k \in m$, by

$$|v_k'^{(i)}| \leq \frac{1}{|\det(A')|} \cdot \|A^{(m+i)}\| \cdot \prod_{j \in m, j \neq k} \|A^{(j)}\| \leq \frac{1}{|\det(A')|} \cdot M^m$$

for $i \in n-m$ and

$$|v_k'^{(n-m)}| \leq \frac{1}{|\det(A')|} \cdot \|d\| \cdot \prod_{j \in m, j \neq k} \|A^{(j)}\| \leq \frac{1}{|\det(A')|} \cdot M^m \,.$$

Since all of the above determinants are integral, it follows that

$$\tilde{v}^{(i)} := \det(A') \cdot v'^{(i)}$$

are integer vectors for $0 \leq i \leq n-m$. Moreover, the vectors $\tilde{v}^{(i)}$, $0 \leq i \leq n-m$, are solutions of

$$A' \cdot x = -\det(A') \cdot A^{(m+i)} \quad \text{and} \quad A' \cdot x = \det(A') \cdot d \,,$$

respectively. We note that $\tilde{v}^{(i)}$ remains a solution of the linear system if we multiply the linear system with a nonzero constant N.

By filling in sufficiently many zeros, the solutions $\tilde{v}^{(i)}$ can be written as vectors in $\mathbb{Z}^{n+1}$, such that

$$(A' \mid A^{(m)} \mid \dots \mid A^{(n-1)} \mid -d) \cdot \underbrace{\begin{pmatrix} \tilde{v}_0^{(0)} & \tilde{v}_0^{(1)} & \cdots & \tilde{v}_0^{(n-m)} \\ \vdots & \vdots & & \vdots \\ \tilde{v}_{m-1}^{(0)} & \tilde{v}_{m-1}^{(1)} & \cdots & \tilde{v}_{m-1}^{(n-m)} \\ \det(A') & 0 & \cdots & 0 \\ 0 & \det(A') & & \\ \vdots & & \ddots & \\ 0 & & & \det(A') \end{pmatrix}}_{=: \tilde{V} \in \mathbb{Z}^{(n+1) \times (n-m+1)}} = 0 \,.$$

The square of the Euclidean norm of the ith-column of $\tilde{V}$, $0 \leq i \leq n-m$, can be bounded by

$$\|\tilde{V}^{(i)}\|^2 \leq m \cdot M^{2m} + \det(A')^2 \leq (m+1) \cdot M^{2m} \,.$$

Multiplying $\tilde{V}$ by the lower part of the generator matrix 7.2.7, i.e. by the rows $m, \dots, m+n$, we get

$$V = \begin{pmatrix} -r_{\max} & 2c_0 & 0 & \cdots & 0 \\ -r_{\max} & 0 & 2c_1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -r_{\max} & 0 & \cdots & \cdots & 2c_{n-1} \\ r_{\max} & 0 & \cdots & \cdots & 0 \end{pmatrix} \cdot \tilde{V} \,.$$

The resulting matrix V is the lower left part of a generator matrix of the lattice of the form 7.2.8. An elementary calculation shows that the norm of column $V^{(i)}$ of V, $0 \le i \le n-m$, can be bounded by

$$\|V^{(i)}\| \le 2\sqrt{n+1} \cdot r_{\max} \cdot \|\tilde{V}^{(i)}\| \le 2\sqrt{(n+1)(m+1)} \cdot r_{\max} \cdot M^m \, .$$

Now, let $b^{(0)}, b^{(1)}, \ldots, b^{(n)}$ be an LLL-reduced basis of a lattice generated by 7.2.7. Using 7.6.15, the length of the first $n-m+1$ basis vectors $b^{(t)}$, $0 \le t \le n-m$, can be bounded by

$$\|b^{(t)}\|^2 \le \left(\frac{4}{4\delta - 1}\right)^n \cdot \max\left\{ \|V^{(i)}\|^2 \mid 0 \le i \le n-m \right\} .$$

If we choose N such that

$$N \ge \left(\frac{4}{4\delta - 1}\right)^{n/2} \cdot 2\sqrt{(m+1)(n+1)} \cdot r_{\max} \cdot M^m \, ,$$

then we have that

$$\|b^{(t)}\| < N$$

for $0 \le t \le n-m$. Thus, the LLL-algorithm will produce a basis whose first $n-m+1$ vectors are all zero in the first m entries, for otherwise the Euclidean length of such a column would be greater than N. Therefore, the LLL-reduced basis has the form 7.2.8. □

For practical purposes it is interesting to note that in almost all cases it suffices to choose N much smaller than the value of the previous bound.

Blockwise Korkine–Zolotarev reduction. As we have seen in Section 7.3, the bounds for the length of the vectors of a Korkine–Zolotarev-reduced basis are much better than the bounds 7.6.14 for an LLL-reduced basis of a lattice L. Unfortunately, no algorithm is known which computes a Korkine–Zolotarev-reduced basis in polynomial time.

In a sense, *Korkine–Zolotarev reduction* is a generalization of the LLL-algorithm. In Step 2 of the LLL-algorithm, we compare the Euclidean length of the projections of $b^{(k)}$ and $b^{(k+1)}$ onto the subspace $\langle b^{(0)}, b^{(1)}, \ldots, b^{(k-1)} \rangle^{\perp}$. In Korkine–Zolotarev reduction, we search for a nontrivial integer linear combination $u_k b^{(k)} + u_{k+1} b^{(k+1)} + \ldots + u_{m-1} b^{(m-1)}$ which minimizes the Euclidean length of

$$\pi_k(u_k b^{(k)} + u_{k+1} b^{(k+1)} + \ldots + u_{m-1} b^{(m-1)}) \, .$$

No algorithm is known which finds the integer linear combination of the shortest nontrivial projection in time which is polynomial in the number of vectors $(m-k)$. Therefore, Schnorr in [172] and [173] restricted the search to blocks of β vectors at a time for some fixed integer constant β. A nontrivial integer linear combination

$$u_k b^{(k)} + u_{k+1} b^{(k+1)} + \ldots + u_{k+\beta-1} b^{(k+\beta-1)}$$

minimizing the Euclidean length of

$$\pi_k(u_k b^{(k)} + u_{k+1} b^{(k+1)} + \ldots + u_{k+\beta-1} b^{(k+\beta-1)})$$

is then found by exhaustive enumeration. This algorithm is called *blockwise Korkine–Zolotarev reduction*. For a further description of improved practical versions, we refer to [173] and [174]. In a blockwise Korkine–Zolotarev-reduced basis of a lattice of rank m the factor $\left(\frac{4}{4\delta-1}\right)^{(m-1)/2}$ in 7.6.14 can be replaced by $(1+\epsilon)^m$ for any fixed $\epsilon > 0$. Of course, the time complexity increases exponentially as ϵ approaches 0.

To summarize the various reduction concepts, the Euclidean norm of the vectors of a reduced basis can be bounded above as follows:

- The LLL-algorithm computes a basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ with
$$\|b^{(t)}\| \leq \left(\frac{4}{4\delta - 1}\right)^{(m-1)/2} \lambda_t(L) \text{ for } t \in m\,.$$
- If $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ is a Korkine–Zolotarev-reduced basis, then
$$\|b^{(t)}\| \leq \left(\frac{t+4}{4}\right)^{1/2} \lambda_t(L) \text{ for } t \in m\,.$$
- If $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ is a blockwise Korkine–Zolotarev-reduced basis, then
$$\|b^{(0)}\| \leq (1+\epsilon)^m \lambda_0(L)\,.$$

Exercises

Exercise Let $\delta \in \mathbb{R}$ with $\frac{1}{4} < \delta < 1$. Show that for $i \in m$ **E.7.6.1**

$$\|\hat{b}^{(i)}\|^2 + \sum_{j \in i} \frac{1}{4} \left(\frac{4}{4\delta - 1}\right)^{i-j} \|\hat{b}^{(i)}\|^2 \leq \left(\frac{4}{4\delta - 1}\right)^{i} \|\hat{b}^{(i)}\|^2\,.$$

Exercise Let $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$ be a sequence of linearly independent vectors in $\mathbb{R}^n$ and $\hat{b}^{(0)}, \hat{b}^{(1)}, \ldots, \hat{b}^{(m-1)}$ the associated Gram–Schmidt vectors. Any vector $v \in \langle b^{(0)}, b^{(1)}, \ldots, b^{(m-1)} \rangle$ can then be written as $v = \sum_{j \in m} r_j b^{(j)} = \sum_{j \in m} r'_j \hat{b}^{(j)}$ with $r_j, r'_j \in \mathbb{R}$ for $j \in m$. Prove the following: If t is the largest index with $r_t \neq 0$, then $r_t = r'_t$. **E.7.6.2**

7.7

7.7 Lattice Point Enumeration

Let us again consider the problem of solving systems of Diophantine equations as described in 7.2.9. Usually, we are interested in finding all solutions to this problem, or to conclude that there are none. In terms of the associated lattice 7.2.7, this mean that we wish to enumerate all lattice points which are subject to a certain set of constraints. Such an approach has first been described by Ritter [170] for $\{0,1\}$ problems. Here we solve the general problem with arbitrary bounds on the variables.

A priori, a lattice $L = \left\{\sum_{i\in m} u_i b^{(i)} \;\middle|\; u_i \in \mathbb{Z}\right\}$ of rank m contains infinitely many elements. It will turn out that there are bounds on the integers $|u_i|$, $i \in m$, which depend solely on the lattice basis $b^{(0)}, b^{(1)}, \ldots, b^{(m-1)}$. These bounds reduce the problem of finding vectors with the properties of 7.2.9 to a finite subset of the original lattice. Therefore we are left with the problem of enumerating all solution vectors of 7.2.9 in a finite subset of the lattice. In the following, we will describe this search process in more detail.

One possibility to compute the above mentioned bounds on the integers $|u_i|$ is by means of Linear Programming. This is not our approach here. Instead, we will use pruning tests to bound the integers $|u_i|$. Compared to the Linear Programming approach, these tests generally lead to a larger enumeration tree. Nevertheless, the pruning tests are very simple and easy to compute, therefore the overall enumeration time seems to be faster than the method based on Linear Programming. The pruning tests we use have quite a long history and are based on the work of [44], [45], [105], [107], [110], [170]. From 7.2.10 we see that a solution vector v, i.e. a vector v of the form 7.2.11, has the upper bounds

7.7.1 $$\|v\|_2^2 \le (n+1)\cdot r_{\max}^2 \quad \text{and}$$

7.7.2 $$\|v\|_\infty \le r_{\max}.$$

The exhaustive enumeration is arranged as a backtracking algorithm starting from $u_{n-m} \in \mathbb{Z}$, which successively chooses values $u_t \in \mathbb{Z}$ for $t = n-m, n-m-1, \ldots, 1, 0$.

7.7.3 **Definition** In each level t of the backtracking algorithm $w^{(t)} = \pi_t(\sum_{j=t}^{n-m} u_j b^{(j)})$ is the projection of the linear combination of the already fixed variables u_t, u_{t+1}, …, u_{n-m} into the subspace of $\mathbb{R}^{n+1}$ which is orthogonal to the linear span $\langle b_0, \ldots, b_{t-1}\rangle$. ◇

Starting from $w^{(n-m+1)} = 0$, $w^{(t)}$ can be iteratively computed from $w^{(t+1)}$ by

7.7.4 $$w^{(t)} = \left(\sum_{i=t}^{n-m} u_i \mu_{it}\right)\hat{b}^{(t)} + w^{(t+1)}\,,$$

with Gram-Schmidt coefficients μ_{it}. In each level t, $n-m \geq t \geq 0$, of the backtrack algorithm we test all possible integer values for the variable u_t. The following tests allow to restrict the possible values of u_t.

- **First pruning condition.** For all $j \leq t$ the vectors $\hat{b}^{(j)}$ are orthogonal to $w^{(t+1)}$ and therefore

  $$\|w^{(t)}\|_2^2 = \left(\sum_{i=t}^{n-m} u_i \mu_{it}\right)^2 \|\hat{b}^{(t)}\|_2^2 + \|w^{(t+1)}\|_2^2 \,.$$

  Further, we notice that $w^{(0)} = \sum_{j=0}^{n-m} u_j b^{(j)}$. Using $\|w^{(j)}\|_2 \geq \|w^{(t)}\|_2$ for $j \leq t$ and 7.7.1 we can backtrack as soon as

  $$\|w^{(t)}\|_2^2 > c := (n+1) \cdot r_{\max}^2 \,.$$

  For fixed $u_{t+1}, \ldots, u_{n-m}$, this gives a bound for u_t:

  $$\left(u_t + \sum_{i=t+1}^{n-m} u_i \mu_{it}\right)^2 \leq \frac{c - \|w^{(t+1)}\|_2^2}{\|\hat{b}^{(t)}\|_2^2} \,.$$

  This is the first pruning condition.

- **Second pruning condition.** Let $\overline{b}^{(i)}, i \in n-m+1$, be a basis of the *dual lattice*, which is defined by the conditions $\langle \overline{b}^{(i)}, b^{(j)} \rangle = \delta_{ij}$ for $0 \leq i,j \leq n-m+1$. If B is the matrix whose columns are the basis vectors $b^{(i)}$, $i \in n-m$, then it was observed in [45] that $u_i = \overline{b}^{(i)\top} \cdot B \cdot u$. Applying the inequality of Cauchy–Schwarz, i.e. $|\overline{b}^{(i)\top} \cdot (B \cdot u)| \leq \|\overline{b}^{(i)}\|_2 \cdot \|B \cdot u\|_2$, for $i \in n-m+1$ then gives the bound

  $$|u_i| \leq \|\overline{b}^{(i)}\|_2 \cdot \|B \cdot u\|_2 \leq \|\overline{b}^{(i)}\|_2 \cdot \sqrt{(n+1) \cdot r_{\max}^2}$$

  and similarly

  $$|u_i| \leq \|\overline{b}^{(i)}\|_1 \cdot r_{\max} \,.$$

  Of course, the numbers $\|\overline{b}^{(i)}\|_1$, $\|\overline{b}^{(i)}\|_2$ can be precomputed before the enumeration.

- **Third pruning condition.** The third test is an adaption to the special situation that we are searching for an integer linear combination of the basis vectors which consists solely of components whose absolute value is bounded by $r_{\max}$. It is based on the following theorem, see [170].

7.7.5 **Theorem** *If the given sequence of integers $u_t, u_{t+1}, \ldots, u_{n-m} \in \mathbb{Z}$ can be extended to $u_0, \ldots, u_t, \ldots, u_{n-m} \in \mathbb{Z}$ such that $\sum_{i \in n-m+1} u_i b^{(i)}$ has the form 7.2.11, then for all $y_t, y_{t+1}, \ldots, y_{n-m} \in \mathbb{R}$:*

$$\left| \sum_{i=t}^{n-m} y_i \|w^{(i)}\|_2^2 \right| \leq r_{\max} \cdot \Big\| \sum_{i=t}^{n-m} y_i w^{(i)} \Big\|_1 .$$

Proof: From 7.7.4 we see that $\langle w^{(l)}, w^{(i)} \rangle = \|w^{(i)}\|_2^2$ for $l < i$. If $w^{(0)}$ has the form 7.2.11 it follows from Hölder's inequality, see exercise 5.1.2, and 7.7.2 that

$$\begin{aligned} \left| \sum_{i=t}^{n-m} y_i \|w^{(i)}\|_2^2 \right| &= \left| \langle w^{(0)}, \sum_{i=t}^{n-m} y_i w^{(i)} \rangle \right| \\ &\leq \|w^{(0)}\|_\infty \cdot \Big\| \sum_{i=t}^{n-m} y_i w^{(i)} \Big\|_1 \\ &\leq r_{\max} \cdot \Big\| \sum_{i=t}^{n-m} y_i w^{(i)} \Big\|_1 . \end{aligned}$$ □

Of course the above inequality can be extended to arbitrary p-norms.

7.7.6 **Remark** We use this theorem in the enumeration algorithm in two ways.

- First, we take $(y_t, y_{t+1}, \ldots, y_{n-m}) = (1, 0, \ldots, 0)$, which results in the test

7.7.7
$$\|w^{(t)}\|_2^2 \leq r_{\max} \|w^{(t)}\|_1 .$$

- Second, we will see that if the test 7.7.7 fails for some vector $w^{(t)} = x\hat{b}^{(t)} + w^{(t+1)}$, then it will also fail for all vectors $\tilde{w}^{(t)} = (x+r)\hat{b}^{(t)} + w^{(t+1)}$ with $r \in \mathbb{Z}$ and $xr > 0$. That means, we can stop the enumeration for these values of $r \in \mathbb{Z}$.

  To show this, let $x \in \mathbb{R}$ and $r \in \mathbb{Z}$ such that $xr > 0$. For $w^{(t)} = x\hat{b}^{(t)} + w^{(t+1)}$ we define $\tilde{w}^{(t)} = (x+r)\hat{b}^{(t)} + w^{(t+1)}$ and we set $\eta := \frac{x}{x+r}$. Then, it is easy to see that

  $$w^{(t)} = \eta \tilde{w}^{(t)} + (1-\eta) w^{(t+1)} \quad \text{and} \quad 0 < \eta < 1.$$

  If $\tilde{w}^{(t)}$ can lead to a solution, then we set $(y_t, y_{t+1}, \ldots, y_{n-m}) = (\eta, 1-\eta, 0, \ldots, 0)$ and get with 7.7.5:

  $$\eta \|\tilde{w}^{(t)}\|_2^2 + (1-\eta) \|w^{(t+1)}\|_2^2 \leq r_{\max} \|\eta \tilde{w}^{(t)} + (1-\eta) w^{(t+1)}\|_1 .$$

  Together, it follows

  $$\begin{aligned} \|w^{(t)}\|_2^2 &\leq \eta \|\tilde{w}^{(t)}\|_2^2 + (1-\eta) \|w^{(t+1)}\|_2^2 \\ &\leq r_{\max} \|\eta \tilde{w}^{(t)} + (1-\eta) w^{(t+1)}\|_1 \\ &= r_{\max} \|w^{(t)}\|_1 . \end{aligned}$$

Therefore, if $\tilde{w}^{(t)}$ can lead to a solution, $w^{(t)}$ can also lead to a solution. On the contrary, if $w^{(t)}$ cannot lead to a solution, i.e. if $\|w^{(t)}\|_2^2 > r_{\max}\|w^{(t)}\|_1$, $\tilde{w}^{(t)}$ cannot lead to a solution for all $\tilde{w}^{(t)} = (x+r)\hat{b}^{(t)} + w^{(t+1)}$ with $r \in \mathbb{Z}$ and $xr > 0$. ◇

Algorithm (Lattice point enumeration) Given the generator matrix 7.2.7 of the lattice $L \subset \mathbb{R}^{m+n+1}$ of rank $n+1$ from 7.2.7 all nonzero vectors $v \in L$ such that $\|v\|_\infty \le r_{\max}$ are determined. 7.7.8

- Compute an LLL-reduced basis $b^{(0)}, b^{(1)}, \dots, b^{(n)}$ of the lattice L.
- Delete the unnecessary columns and rows of the generator matrix according to Section 7.2. The remaining basis $b^{(0)}, b^{(1)}, \dots, b^{(n-m)} \subset \mathbb{R}^{n+1}$ has rank $n-m+1$.
- Compute the Gram–Schmidt vectors $\hat{b}^{(0)}, \hat{b}^{(1)}, \dots, \hat{b}^{(n-m)}$ together with the Gram–Schmidt coefficients μ_{ij}, see 7.4.2.
- Set $R := (n+1) \cdot r_{\max}^2$.
- The recursive backtracking algorithm enum() has two input parameters. The first parameter t is the search level, it runs from $n-m$ down to 0. The second parameter $w' \subset \mathbb{R}^{n+1}$ is the vector which has been computed in the level $t+1$. The enumeration is initiated with the call of enum($n-m$, 0).

```
(1)  function enum($t, w'$)
(2)  begin
(3)    firstprune := false
(4)    $y_t := \sum_{i=t+1}^{n-m} u_i\mu_{it}$
(5)    $u_t := \lfloor -y_t \rceil$
(6)    while true
(7)      $w := \big(\sum_{i=t}^{n-m} u_i\mu_{it}\big)\hat{b}^{(t)} + w'$
(8)      if $\|w\|^2 > R$ then return               /* step back */
(9)      if $t > 0$ then
(10)       if prune($u_t$) then
(11)         if firstprune then return          /* step back */
(12)         else
(13)           next($u_t$)
(14)           firstprune := true
(15)           goto line (7)
(16)         end if
(17)       else
(18)         enum($t-1, w$)                     /* step forward */
```

```
(19)       else                                   /* $t = 0 \rightarrow$ solution */
(20)          if $w$ has the form 7.2.11 then print $w$
(21)       next($u_t$)
(22)    end while
(23) end                                                                  □
```

The procedure next() in lines (13) and (21) determines the next possible integer value of the variable u_t. Initially, when entering a new level t, in line (5) u_t is set to be the closest integer value of $-y_t := -\sum_{i=t+1}^{n-m} u_i \mu_{it}$, say u_t^1. The next value u_t^2 of u_t is the second closest integer to $-y_t$ then follows u_t^3 and so forth. Therefore the values of u_t alternate around $-y_t$. If the function prune() returns true for w_t, then we do one more regular call of the procedure next() in line (13), i.e. u_t is set to be the next closest integer to $-y_t$. In Fig. 7.7 this happens while u_t^4 is determined.

After that, using 7.7.6, the enumeration proceeds only in this remaining direction. Compare the computation of u_t^5 in Fig. 7.7. Finally, the second time when the function prune() returns true, the algorithm steps back and increases the enumeration level, see line (11).

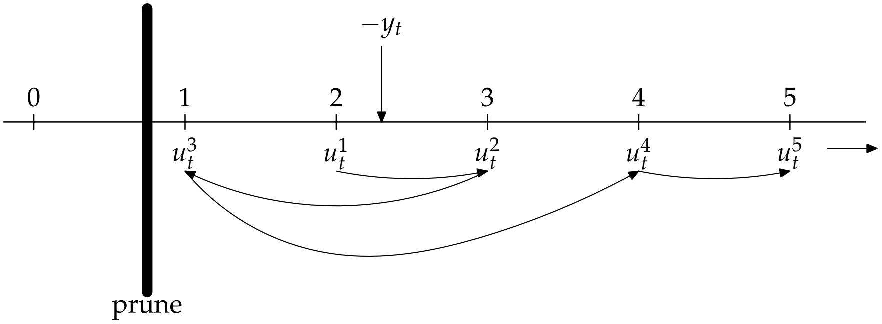


Fig. 7.7 Enumeration in level t and pruning after u_t^3

7.7.9 **Example** Suppose $y_t = -2.3$ for $0 \leq t \leq n-m$. Therefore, $-y_t = 2.3$ and according to line (5), $u_t = \text{round}(-y_t) = \lfloor 2.3 + \frac{1}{2} \rfloor = 2$. First, assume that the procedure prune() always return false. Therefore, in the subsequent calls of the procedure next() in line (21), the variable u_t takes the values $3, 1, 4, 0, 5, -1, 6, \ldots$.

Now assume, after testing the values $u_t = 2$ and 3, that the procedure prune() returns true for $u_t = 1$. Then the value of u_t is set to 4 in line (13).

After subsequent calls of next() in line (21), u_t takes the values $5, 6, 7$ and so forth until prune() returns true the next time. ◇

The function prune for the third pruning test according to 7.7.7 can be implemented as follows.

Algorithm 7.7.10

```
function prune($w_t$)
    if $\|w^{(t)}\|^2 \leq r_{\max} \cdot \|w^{(t)}\|_1$
        return false
    else
        return true
    end if
```

□

The first pruning test from page 599 is done in line (8) in 7.7.8, the second pruning test from page 599 can be additionally added after line (6) in 7.7.8.

If in the forward step of the algorithm, i.e. in line (18) a new level t is entered, then initially in the next call of enum() in line (5) u_t is set to the closest integer to $-\sum_{i=t+1}^{n-m} u_i \mu_{it}$. Since

$$\|w^{(t)}\|_2^2 = \left(u_t + \sum_{i=t+1}^{n-m} u_i \mu_{it}\right)^2 \|\hat{b}^{(t)}\|_2^2 + \|w^{(t+1)}\|_2^2,$$

this choice of u_t minimizes $\|w^{(t)}\|_2^2$.

Example We illustrate the algorithm by solving a system of Diophantine linear equations which occurs during the construction of linear codes with prescribed minimum distance in Chapter 8, Example 8.4.4. In order to find a $(14, 3, 9)$-code over $\mathbb{F}_3$ the following system must be solved: 7.7.11

$$\begin{pmatrix} 2 & 2 & 0 & -1 & 0 & 0 \\ 1 & 1 & 2 & 0 & -1 & 0 \\ 0 & 3 & 1 & 0 & 0 & -1 \\ 3 & 6 & 4 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 14 \end{pmatrix},$$

where $x_0 \in \{0, 1, 2, 3, 4\}$, $x_1 \in \{0, 1, 2\}$, $x_2 \in \{0, 1, 2, 3\}$ and $x_i \in \{0, 1, 2, 3, 4, 5\}$ for $i \in \{3, 4, 5\}$. According to 7.2.7, the lattice is generated by the matrix

$$\left(\begin{array}{c|cccccc}
0 & 2000 & 2000 & 0 & -1000 & 0 & 0 \\
0 & 1000 & 1000 & 2000 & 0 & -1000 & 0 \\
0 & 0 & 3000 & 1000 & 0 & 0 & -1000 \\
-14000 & 3000 & 6000 & 4000 & 0 & 0 & 0 \\
\hline
-60 & 30 & 0 & 0 & 0 & 0 & 0 \\
-60 & 0 & 60 & 0 & 0 & 0 & 0 \\
-60 & 0 & 0 & 40 & 0 & 0 & 0 \\
-60 & 0 & 0 & 0 & 24 & 0 & 0 \\
-60 & 0 & 0 & 0 & 0 & 24 & 0 \\
-60 & 0 & 0 & 0 & 0 & 0 & 24 \\
\hline
60 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

where $r_{\max} = \operatorname{lcm}(4,2,3,5) = 60$ and the constant N is set to $N = 1000$. In the first step of the algorithm, the LLL-reduction is applied to the columns of the above matrix. This results in the following new basis:

$$\left(\begin{array}{ccc|cccc}
0 & 0 & 0 & -1000 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1000 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1000 \\
0 & 0 & 0 & 0 & 0 & 1000 & 0 \\
\hline
-60 & -60 & 0 & 0 & 0 & -30 & 0 \\
60 & 0 & -60 & 0 & 0 & 0 & 0 \\
0 & 20 & 100 & 0 & 0 & 40 & 0 \\
-48 & -12 & -132 & 24 & 0 & -48 & 0 \\
-24 & 60 & 12 & 0 & 24 & 24 & 0 \\
72 & 60 & -60 & 0 & 0 & 24 & -24 \\
\hline
0 & 60 & -60 & 0 & 0 & 0 & 0
\end{array}\right) .$$

The first three columns correspond to solutions of the above system. But the first column corresponds to a solution where the right hand side of the above system is not included since the last entry is equal to zero. The second column corresponds to a solution, because all entries have absolute value at most 60. The solution $x = (x_1, x_2, \ldots, x_6)^\top$ of the original system of equations can now be obtained by solving

$$\begin{pmatrix} -60 \\ 0 \\ 20 \\ -12 \\ 60 \\ 60 \\ 60 \end{pmatrix} = \begin{pmatrix}
-60 & 30 & 0 & 0 & 0 & 0 & 0 \\
-60 & 0 & 60 & 0 & 0 & 0 & 0 \\
-60 & 0 & 0 & 40 & 0 & 0 & 0 \\
-60 & 0 & 0 & 0 & 24 & 0 & 0 \\
-60 & 0 & 0 & 0 & 0 & 24 & 0 \\
-60 & 0 & 0 & 0 & 0 & 0 & 24 \\
60 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} ,$$

which results in $x = (0,1,2,2,5,5)^\top$. For the exhaustive enumeration of all solutions we can remove the unnecessary rows and columns and use the lattice which is generated by the matrix

$$\left(\begin{array}{rrr} -60 & -60 & 0 \\ 60 & 0 & -60 \\ 0 & 20 & 100 \\ -48 & -12 & -132 \\ -24 & 60 & 12 \\ 72 & 60 & -60 \\ \hline 0 & 60 & -60 \end{array}\right).$$

Eventually, after 6 loops in the exhaustive enumeration step it is determined that there are no further solutions. ◇

Exercises

E.7.7.1 **Exercise** Use the computer software on the CD-ROM of the book to compute the solution of the system of Diophantine linear equations from 7.7.11.

7.8 Computing the Minimum Distance of Linear Codes

7.8

Let C be a binary or ternary linear code. It is possible to compute the minimum distance of such a code by using a variant of the lattice point enumeration algorithm from Section 7.7. For this purpose, we note that in the binary case we have $-1 \equiv 1 \bmod 2$ while in the ternary case $-1 \equiv 2 \bmod 3$. Thus, codewords of binary or ternary codes can be represented by vectors with integral entries in $\{0,1,-1\}$.

Let $\mathbb{F}$ be a binary or ternary field, i.e. $q = 2$ or $q = 3$. Consider the lattice L_C which is spanned by the columns of the integral $(n+k) \times (k+n)$-matrix

$$B_C = \left(\begin{array}{c|c} N \cdot \Gamma^\top & N \cdot qI_n \\ \hline I_k & 0 \end{array}\right),$$

where Γ is a $k \times n$ generator matrix of the code C and N is a large integer constant. The matrix qI_n is used to reduce the integral linear combinations of the columns of $\Gamma^\top$ modulo q. Any lattice vector $v \in L_C$ with $v_i \in \{0,1,-1\}$ for $i \in n$ corresponds to a codeword $v_C \in C$ and $\mathrm{wt}(v_C)$ is the number of nonzero entries in the first n coefficients of v. Thus, the minimum distance problem can be solved by finding a nonzero lattice vector with the least number (> 0) of nonzero entries in the first n rows.

If the constant N is large enough, the reduced lattice basis contains k vectors whose first n entries are all zero. These vectors can be removed. Further, the lower k components are no longer necessary and can be removed, too. To achieve an even better reduced basis, a useful strategy is to shuffle the remaining basis vectors randomly and apply lattice basis reduction to the reordered basis. This mixing and reduction step can be repeated several times. Finally, the resulting basis is enumerated with 7.7.8, as described below. Here is an example.

7.8.1 **Example** Since we write codewords as row vectors, we apply lattice basis reduction to rows in this example. So, the basis vectors are the rows of the generator matrix of the lattice L_C.

The goal is to determine the minimum distance of the ternary Golay code. It is also a quadratic-residue-code $C_Q(11,6)$ over $\mathbb{F}_3$, see Section 4.4. A generator matrix is

$$\Gamma = \begin{pmatrix} 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 \end{pmatrix}.$$

Using $N = 6$, the generator matrix $B_C^\top$ of the lattice L_C is

$$B_C^\top = \left(\begin{array}{ccccccccccc|cccccc} 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right).$$

Applying the LLL-algorithm to the *rows* of $B_C^\top$ gives

$$B_C = \left(\begin{array}{rrrrrrrrrrr|rrrrrr}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 \\
\hline
0 & -6 & -6 & -6 & 0 & -6 & -6 & 0 & -6 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -6 & -6 & 6 & -6 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 \\
-6 & 0 & -6 & 0 & 0 & 0 & -6 & -6 & -6 & 0 & -6 & 1 & -1 & 0 & 1 & 0 & -1 \\
0 & -6 & -6 & 0 & 0 & -6 & 0 & 0 & 0 & 6 & 6 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & -6 & 0 & 6 & 0 & -6 & 0 & -6 & 6 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \\
0 & -6 & 0 & 0 & -6 & 0 & -6 & -6 & -6 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 \\
0 & 0 & 6 & 0 & 6 & 6 & 6 & 0 & 0 & 0 & 6 & 0 & 0 & -1 & 1 & 0 & 1 \\
0 & 0 & 6 & 0 & 0 & 6 & 0 & 6 & 6 & 6 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\
-6 & -6 & 0 & 0 & 0 & -6 & 0 & 0 & -6 & 0 & -6 & 1 & 0 & 1 & 1 & 0 & -1 \\
-6 & 6 & 0 & 0 & -6 & 6 & 6 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
-6 & -6 & -6 & 0 & 0 & 0 & -6 & 0 & 0 & -6 & 0 & 1 & 0 & -1 & 0 & -1 & 0
\end{array}\right).$$

We delete the unnecessary rows and columns, see 7.2.8. Then scaling and mixing the remaining rows gives

$$\begin{pmatrix}
0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
-1 & 0 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 \\
-1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 \\
-1 & 1 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 0 & 1 \\
0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 \\
0 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & -1 & 1 & 0 \\
-1 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0
\end{pmatrix}.$$

LLL-reduction of this lattice produces the following basis

$$\begin{pmatrix}
0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 \\
-1 & 1 & 0 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 \\
0 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 0 & 1 \\
-1 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 \\
0 & -1 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 1 & 0 \\
-1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\
0 & 0 & 1 & 1 & -1 & 1 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & -1 & 0
\end{pmatrix}.$$

The first row corresponds to a codeword of weight 5:

$$v = (0, -1, -1, 0, 0, -1, 0, 0, 0, 1, 1).$$

After 57 executions of the loop, algorithm 7.7.8 with the improvements described below determines that there exists no nonzero vector with weight ≤ 4. That means the minimum distance of the ternary Golay code is equal to $d = 5$.

◇

If we use a systematic generator matrix of C of the form $\Gamma = (I_k \mid A)$, $A \in \mathbb{F}_q^{k\times n-k}$, we can do even better. We use the lattice L_C which is generated by

$$B_C = \left(\begin{array}{c|c} A^\top & qI_{n-k} \\ \hline I_k & 0 \end{array}\right),$$

It has the advantage that the constant N is not longer needed. In order to find a nonzero codeword of minimal weight, we have to find a nonzero lattice vector v in the rank n lattice $L_C \subseteq \mathbb{Z}^n$ with $\|v\|_\infty = 1$ which contains the minimal number of nonzero entries. Note that if $\mathrm{wt}(v) = s$ and $\|v\|_\infty = 1$, then also $\|v\|_2^2 = s$.

The minimum distance of C can be computed by a variation of 7.7.8. Initially, in 7.7.8 we set $R = d - 1$, where d is an upper bound for the minimum distance of C. If no other bound is known, d is the weight of the shortest codeword in the generator matrix.

Then, the backtracking of the lattice point enumeration algorithm as described in 7.7.8 is started. If a lattice vector $v \in L_C$ with $\|v\|_\infty = 1$ and $\|v\|_2^2 \leq R$ is found during the enumeration then it is printed, after line (24) R is set to $R := \|v\|_2 - 1$, and the backtracking is continued. If it is known that the minimum distance of C is a multiple of some integer c – for example if C is a doubly even code – then we can even set $R := \|v\|_2 - c$ in this situation.

Further improvements in the enumeration can be obtained by modifying the lattice point enumeration of Section 7.7. For an integer $0 < t < n$ and a vector $v \in \mathbb{R}^n$, we define

$$\max_t(v)$$

to be the sum of the t largest absolute values of entries of v. For example, if $v = (-1, 2.5, -3, 0.5)^\top$, then $\max_2(v) = 3 + 2.5 = 5.5$.

Let $R = d - 1$, where $d > 1$ is an upper bound on the minimum distance of the code C and let $b^{(0)}, b^{(1)}, \ldots, b^{(n-1)}$ be a basis of the lattice L_C. With the notation of Section 7.7, 7.7.5 can be adapted to the computation of the minimum distance of a linear code in the following way.

7.8.2 **Theorem** *Let $t \in n$. If for fixed $u_t, u_{t+1}, \ldots, u_{n-1} \in \mathbb{Z}$ there exist coefficients u_0, $u_1, \ldots, u_{t-1} \in \mathbb{Z}$ with $\|\sum_{i\in n} u_i b^{(i)}\|_\infty \leq 1$ and $\|\sum_{i\in n} u_i b^{(i)}\|_2^2 \leq R$, then for all y_t, $y_{t+1}, \ldots, y_{n-1} \in \mathbb{R}$:*

7.8.3
$$\left|\sum_{i=t}^{n-1} y_i \|w^{(i)}\|_2^2\right| \leq \max_R\left(\sum_{i=t}^{n-1} y_i w^{(i)}\right).$$

Proof: We have $\langle w^{(l)}, w^{(i)}\rangle = \langle w^{(i)}, w^{(i)}\rangle$ for $0 \le l < i < n$. If there exist $u_0, u_1, \ldots, u_{n-1} \in \mathbb{Z}$ such that for $w^{(0)} = \sum_{i\in n} u_i b^{(i)}$ simultaneously

$$\|w^{(0)}\|_\infty = 1 \text{ and } \|w^{(0)}\|_2^2 \le R,$$

then it is easy to see that for an arbitrary vector $v \in \mathbb{R}^n$ the inequality

$$|\langle w^{(0)}, v\rangle| \le \max_R(v)$$

holds. It follows that

$$\begin{aligned}\left|\sum_{i=t}^{n-1} y_i \langle w^{(i)}, w^{(i)}\rangle\right| &= \left|\sum_{i=t}^{n-1} y_i \langle w^{(0)}, w^{(i)}\rangle\right| \\ &= \left|\langle w^{(0)}, \sum_{i=t}^{n-1} y_i w^{(i)}\rangle\right| \\ &\le \max_R\Big(\sum_{i=t}^{n-1} y_i w^{(i)}\Big).\end{aligned}$$

□

Therefore, during the computation of the minimum distance of linear codes we can replace in the enumeration algorithm 7.7.8 the test in 7.7.5 by 7.8.3. Experiments show that 7.7.8 together with 7.8.3 can determine the minimum distance of quadratic-residue-codes for values of n at least up to 100.

Example For the quadratic-residue-code $C_Q(37, 19)$ over $\mathbb{F}_3$, whose generator matrix is generated cyclically by the vector **7.8.4**

(1,1,2,2,1,2,2,0,2,2,2,0,2,2,1,2,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),

the LLL-algorithm determines the vector

(0,0,0,1,0,0,0,1,0,1,1,0,1,0,0,0,0,0,1,2,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0)

of weight 10. The enumeration 7.7.8 together with 7.8.3 needs 586 799 iterations to show that there is no codeword of lower weight. The parity extension of $C_Q(37, 19)$ has minimum weight 11. A vector with minimum weight is

(1,2,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,1,2,0,0,0,0,2,0,2,0,0,0,0,2,1,0,0,0,0,0,2). ◇

Example The generator matrix of the quadratic-residue-code $C_Q(61, 31)$ over $\mathbb{F}_3$ is generated by the vector **7.8.5**

(1,0,2,1,2,2,0,0,0,1,0,2,1,1,2,1,2,1,1,2,0,1,0,0,0,2,2,1,2,0,
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0).

The LLL-algorithm computes the vector

(0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,2,1,0,
0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,2,0,2,2,0,0,0,0,0,2,0,0,1,0,0,0)

which has weight 11. The exhaustive enumeration determines that there is no vector of lower weight. The parity extension of $C_Q(61,31)$ has minimum weight 12. A vector with minimum weight is

(2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,2,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,1,2,0,0,1,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,1,0,0,0). ◇

7.8.6 **Example** The generator matrix of the quadratic-residue-code $C_Q(71,36)$ over $\mathbb{F}_3$ is generated by the vector

(2,2,2,2,2,2,2,0,0,2,2,1,1,2,2,2,0,2,2,2,0,1,1,0,2,0,2,0,1,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0).

The LLL-algorithm computes

(0,0,0,0,0,1,0,1,0,0,1,1,1,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,
2,2,0,0,0,0,0,0,0,0,0,2,0,0,0,0,2,1,0,0,0,0,2,0,2,0,2,0,0,2,0,0,0,0,0)

of weight 17. The parity extension of $C_Q(71,36)$ has 18 as upper bound for minimum distance. A vector attaining this bound is

(0,0,0,1,0,0,0,1,1,1,2,1,0,0,2,0,0,1,0,2,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,
1,0,0,0,2,0,0,0,0,1,0,0,1,0,0,0,1,0,0,2,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0). ◇

7.8.7 **Example** The generator matrix of the quadratic-residue-code $C_Q(83,42)$ over $\mathbb{F}_3$ is generated by

(2,0,1,2,1,2,0,0,1,2,2,0,0,2,1,2,1,1,0,0,1,1,1,1,0,0,2,2,1,1,2,1,0,2,0,1,0,0,1,2,1,
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0).

The LLL-algorithm computes

(0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,2,0,0,2,0,1,2,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,
0,2,0,0,0,0,0,0,2,0,0,1,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,2,0,2,0,1,0,2,0,0,1,1,0,0,0,0,0).

Therefore, an upper bound for the minimum distance is 20. It follows that the parity extension of $C_Q(83,42)$ has 21 as upper bound for the minimum distance.

The generator matrix of the quadratic-residue-code $C_Q(97,49)$ over $\mathbb{F}_3$ is generated by the vector

(1,1,1,0,0,1,1,0,2,1,0,0,0,1,0,2,1,2,0,0,2,2,2,0,1,0,2,2,2,0,0,2,1,2,0,1,0,0,0,1,2,0,1,1,0,0,1,1,
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0).

The LLL-algorithm finds

$$\begin{gathered}(0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,0,0,0,2,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,2,2,0,2,0,1,0,0,0,\\ 0,0,2,0,2,1,1,0,0,0,0,1,0,0,0,2,0,1,1,1,0,0,2,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,0,0,2,0,1,0,0,0).\end{gathered}$$

Therefore, an upper bound for the minimum distance is 23. ◇

If the above examples are reproduced with the software from the enclosed CD-ROM the advantages and disadvantages of this algorithm can be seen. The LLL-algorithm is very good in computing codewords of small weight very fast. The second phase, which deterministically computes a codeword with minimum weight and proves that there are no codewords of smaller weight still needs exponential time.

Exercises

Exercise Use the computer software on the CD-ROM of the book to compute the minimum distance of the binary Golay code with generator matrix **E.7.8.1**

$$G = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&0&0&0&1&1\\
0&1&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&0&1&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&1&0&1&0&1&1\\
0&0&0&1&0&0&0&0&0&0&0&0&1&1&0&0&0&1&1&1&0&1&1&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&0&0&1\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&0&1\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&0&1&1&0&1&1&1\\
0&0&0&0&0&0&0&1&0&0&0&0&1&0&1&1&0&1&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&1&1&0&1&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&1&1&0&1&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&0&0&0&1&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&1&1&0&0&0&1&1&1
\end{pmatrix}.$$

Compute the minimum distance of this code over $\mathbb{F}_3$.