Lattices and the LLL Algorithm

Lattices are beautiful mathematical objects with applications all over mathematics and theoretical computer science. Examples include:

- **Sphere Packing:** A classical problem called the "Sphere Packing Problem" asks for a way to pack the largest number of spheres of equal volume in 3-dimensional space (in an asymptotic sense, as the volume of the available space goes to infinity). The so-called *Kepler's Conjecture*, turned into a theorem by Hales, states that the face-centered cubic lattice offers the optimal packing of spheres in 3 dimensions.

  Higher dimensional settings form a fascinating question in the geometry of numbers (the study of lattices). The dimension-8 and 24 cases were recently resolved in exciting works by Viazovska showing that the so-called E_8 lattice provides the optimal packing in dimension 8, and by Cohn, Kumar, Miller, Radchenko and Viazowska showing that the so-called Leech lattice provides the optimal packing in dimension 24.

  How about other dimensions? This remains a mystery.

- **Error Correcting Codes:** Generalizing to n dimensions, the sphere packing problem and friends have applications to constructing *error-correcting codes* with the optimal rate.

- **Number Theory:** In mathematics, the study of lattices is called the "Geometry of Numbers", a term coined by Hermann Minkowski. Minkowski's Theorem and subsequent developments have had an enormous impact on Number Theory, Functional Analysis and Convex Geometry. Lattices have been used to test various number theoretic conjectures, the most famous being a disproof of Merten's Conjecture by Odlyzko and te Riele in 1985.

Lattices have also been quite influential in Theoretical Computer Science:

- In **Algorithms:** The famed Lenstra-Lenstra-Lovász (LLL) algorithm for the shortest vector problem that we will see today has generated a treasure-trove of algorithmic applications. Lattices have been used to construct an Integer Linear Programming algorithm in constant dimensions, in factoring polynomials over the rationals, and algorithms to find small solutions to systems of polynomial equations.

- In **Complexity Theory:** Lattices provide one of the most striking sources of problems with a worst-case to average-case connection. NP-hard problems are widely believed to be hard in the worst case, but are they hard on typical or "average" instances? (Note that these terms have to be defined precisely: by endowing the input space with a probability distribution that defines what "typical" or "average" means.) For many problems and many average-case distributions, we know that this is not the case. In contrast, for the (approximate) shortest vector problem, we can show that finding a solution in a "random lattice" chosen from a certain easily sampleable distribution is as hard as finding a solution in the worst case, namely for arbitrary lattices.

- In **Cryptography:** The first applications of lattices in Cryptography have been in breaking cryptosystems, for example, variants of the knapsack cryptosystem, the NTRU cryptosystem and special cases of the RSA function. More recently, however, lattices have been used quite successfully in constructing secure cryptographic algorithms that achieve highly expressive functionalities such as fully homomorphic encryption.

In this course, we will study lattices from the point of view of theoretical computer science, first the mathematics of lattices, then the algorithms and complexity theory and finally lattice-based cryptography.

Notation. We will denote the natural numbers by $\mathbb{N}$, integers by $\mathbb{Z}$, rationals by $\mathbb{Q}$ and the reals by $\mathbb{R}$.

1 Lattices

Throughout, we treat all vectors as column vectors unless otherwise specified. For a matrix $\mathbf{B}$ (resp. row vector $\mathbf{v}$), $\mathbf{B}^T$ (resp. $\mathbf{v}^T$) denotes the transpose of $\mathbf{B}$ (resp. $\mathbf{v}$).

Definition 1 (Lattices)**.** *Given n* linearly independent *vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice generated by them is defined as*

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \right\}$$

We call $\mathbf{b}_1, \ldots, \mathbf{b}_n$ a *basis* of the lattice. Note that the definition requires $\mathbf{b}_1, \ldots, \mathbf{b}_n$ to be linearly independent over $\mathbb{R}$ (and not over $\mathbb{Z}$). For example, the two vectors $(1 \;\; 0)^T$ and $(\sqrt{2} \;\; 0)^T$ do not form a basis for a lattice according to this definition since they are not linearly independent over $\mathbb{R}$, even though they are linearly independent over $\mathbb{Z}$.

We call n the *rank* of the lattice, and m the *dimension* of the lattice. In general, $n \leq m$. When $n = m$, we call the lattice a *full-rank* lattice. Throughout this course, we will focus on full-rank lattices although most results we prove can be generalized to the non full-rank case.

We will use a notational short-hand when dealing with bases, letting a matrix $\mathbf{B}$ whose columns are the basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ denote a lattice basis. That is, we will write

$$\mathbf{B} = \begin{pmatrix} | & & | \\ \mathbf{b}_1 & \ldots & \mathbf{b}_n \\ | & & | \end{pmatrix}$$

and thus, in this notation,

$$\mathcal{L}(\mathbf{B}) \stackrel{\text{def}}{=} \{\mathbf{B}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\}$$

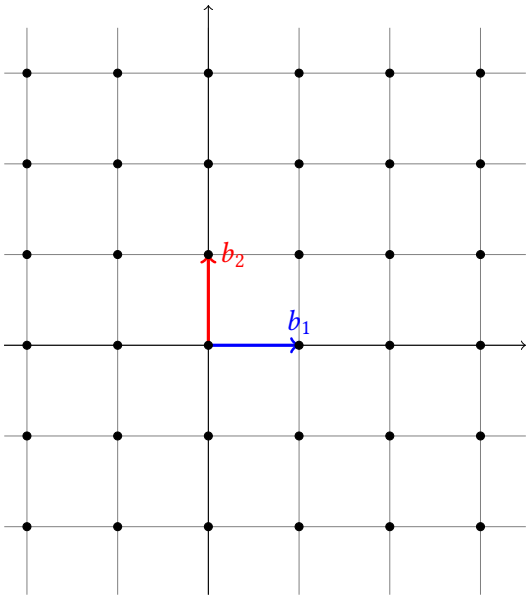
Examples of Lattices.

1. Figure 1(a) shows the lattice in 2 dimensions generated by the vectors $(1, 0)^T$ and $(0, 1)^T$. This lattice is the set of all points in $\mathbb{R}^2$ with integer coordinates.

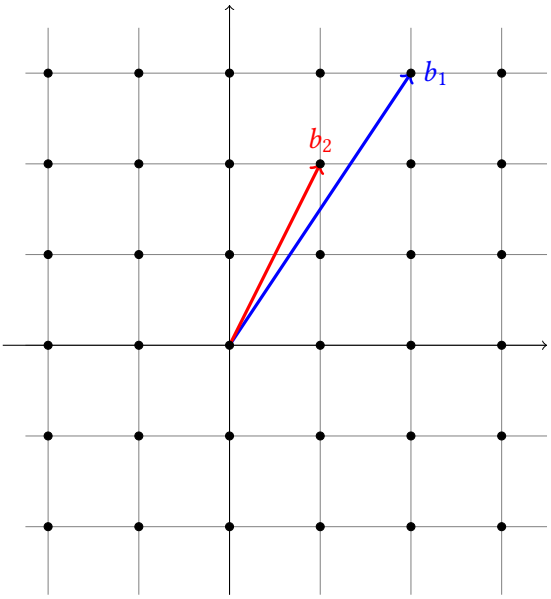
   This can be generalized to n dimensions, where the lattice $\mathbb{Z}^n$ is called *the integer lattice*.

2. Figure 1(b) shows a different basis for the same lattice, namely the basis consisting of the vectors $(1, 2)^T$ and $(2, 3)^T$.

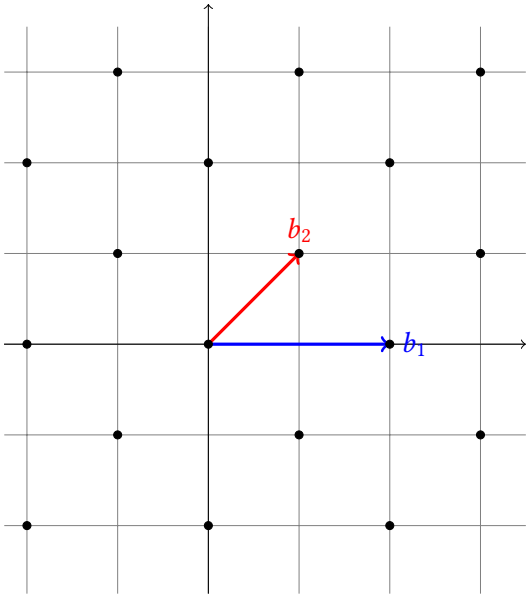
3. Figure 1(c) shows a different lattice in 2 dimensions, generated by the basis vectors $(2, 0)^T$ and $(1, 1)^T$. Note that this is a sub-lattice of $\mathbb{Z}^2$, namely a subset of $\mathbb{Z}^2$ which is also a lattice. (We will formally define sublattices later in the course).



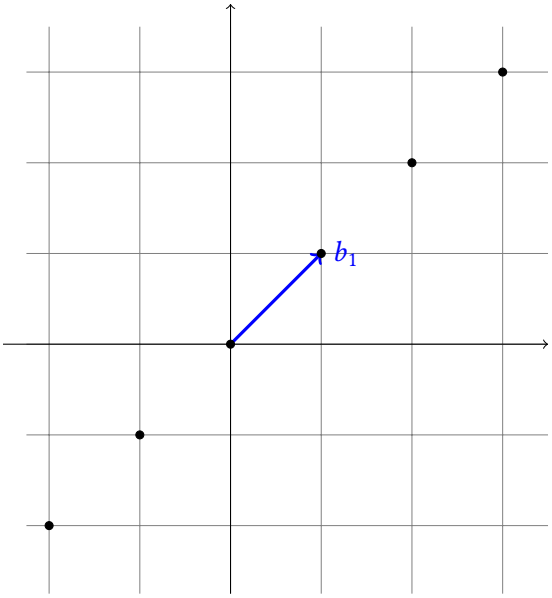
(a) The lattice $\mathbb{Z}^2$ with basis vectors $(0, 1)$ and $(1, 0)$.



(b) The lattice $\mathbb{Z}^2$ with a different basis consisting of vectors $(1, 2)$ and $(2, 3)$. In fact, any lattice has infinitely many bases.



(c) A full-rank lattice generated by the basis vectors $(1, 1)$ and $(2, 0)$. Note that this is a sub-lattice of $\mathbb{Z}^2$.



(d) A *non full-rank* lattice with basis vector $(1, 1)$

Figure 1: Various lattices and their bases.

4. In one dimension, all lattices are multiples of a single number. For example, the lattice generated by (2) is the set of all even numbers.

5. All the examples we saw so far are full-rank lattices. Figure 1(d) shows a lattice in 2 dimensions generated by the vector $(1,1)^T$ – this lattice has rank 1. We will not deal with non full-rank lattices in this course.

6. The set of points generated by (1) and $(\sqrt{2})$ in one dimension is not a lattice. First, this example does not conform to Definition 1 since 1 and $\sqrt{2}$ are *linearly dependent* over $\mathbb{R}$.

   Secondly, it turns out that any n-dimensional lattice is a discrete subgroup of $\mathbb{Z}^n$. However, the set generated by (1) and $(\sqrt{2})$ is not a discrete subset of $\mathbb{Z}$ since one can generate arbitrarily small numbers as linear combinations of 1 and $\sqrt{2}$.

It is instructive to compare the definition of a lattice generated by n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ to the definition of the span of these vectors.

Definition 2 (Span)**.** *Given n* linearly independent *vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, their* span *is defined as*

$$\mathsf{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{R} \right\}$$

Note the difference between Definition 1 of a lattice generated by a set of vectors – which consists of all of its *integer* linear combinations – and the above definition of the span of a set of vectors – which consists of all of its linear combinations with *real* coefficients. The crucial power of lattices comes from the fact that it is a discrete set. Clearly, $\mathsf{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \supset \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$.

2 Same Lattice, Many Bases

We already saw from the examples above (Figure 1(a) and Figure 1(b)) that the same lattice can have many different bases. For example, it turns out that all the bases given below generate the same lattice, namely $\mathbb{Z}^2$:

$$\mathbf{B}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{B}_2 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{B}_3 = \begin{pmatrix} 647 & 64 \\ 91 & 9 \end{pmatrix}$$

but the following basis *does not* generate $\mathbb{Z}^2$, but only a proper *sub-lattice* of $\mathbb{Z}^2$.

$$\mathbf{B}_4 = \begin{pmatrix} 42 & 41 \\ 9 & 8 \end{pmatrix}$$

In fact, any lattice has infinitely many bases. In particular, the bases can have arbitrarily large coefficients.

A natural question to ask is: *how can we efficiently tell if two given bases $\mathbf{B}$ and $\mathbf{B}'$ generate the same lattice?* We will give two answers to this question – an algebraic answer and a geometric answer.

2.1 An Algebraic Characterization using Unimodular Matrices

Our first characterization provides an efficient algorithm to determine if two bases generate the same lattice. In order to present the characterization, we first need to define the notion of a *unimodular matrix*.

Notation. For any $x \in \mathbb{R}$, we will let $|x|$ represent the absolute value of x.

Definition 3. *A matrix* $\mathbf{U} \in \mathbb{Z}^{n \times n}$ *is unimodular if* $|\det(\mathbf{U})| = 1$.

Here, $\det(\mathbf{U})$ denotes the determinant of the (square) matrix $\mathbf{U}$, and $|\cdot|$ denotes the absolute value. For example, the matrix $\left(\begin{smallmatrix} 2 & 1 \\ 1 & 1 \end{smallmatrix}\right)$ is unimodular, and so is $\left(\begin{smallmatrix} 647 & 64 \\ 91 & 9 \end{smallmatrix}\right)$, but not $\left(\begin{smallmatrix} 42 & 41 \\ 9 & 8 \end{smallmatrix}\right)$.

Proposition 4. *If* $\mathbf{U}$ *is unimodular, so is* $\mathbf{U}^{-1}$.

We can now state the characterization of equivalent bases.

Theorem 5. *Given two full-rank bases* $\mathbf{B} \in \mathbb{R}^{n \times n}$ *and* $\mathbf{B}' \in \mathbb{R}^{n \times n}$*, the following two conditions are equivalent:*

- $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$
- *There exists a unimodular matrix* $\mathbf{U}$ *such that* $\mathbf{B}' = \mathbf{B}\mathbf{U}$.

Proof. ("⇒") First, assume that $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$. Then, there are integer matrices $\mathbf{V}$ and $\mathbf{V}'$ such that

$$\mathbf{B}' = \mathbf{B}\mathbf{V} \qquad \text{and} \qquad \mathbf{B} = \mathbf{B}'\mathbf{V}'$$

It suffices to show that $|\det(\mathbf{V})| = |\det(\mathbf{V}')| = 1$.

Putting these two equations together, we have $\mathbf{B}' = \mathbf{B}\mathbf{V} = \mathbf{B}'(\mathbf{V}'\mathbf{V})$. Since $\mathbf{B}'$ is non-singular (remember: $\mathbf{B}$ is a full-rank matrix, and so is $\mathbf{B}'$) we can multiply both sides of the equation by $(\mathbf{B}')^{-1}$ and we get

$$\mathbf{V}'\mathbf{V} = \mathbf{1}_n \tag{1}$$

were $\mathbf{1}_n$ denotes the n-by-n identity matrix.

Since determinant is multiplicative, we get $\det(\mathbf{V}')\det(\mathbf{V}) = 1$. Since $\mathbf{V}$ and $\mathbf{V}'$ are integer matrices, their determinant is also an integer.

Putting these two facts together, we see that the only two choices are:

- $\det(\mathbf{V}) = \det(\mathbf{V}') = 1$, or
- $\det(\mathbf{V}) = \det(\mathbf{V}') = -1$

In either case, $|\det(\mathbf{V})| = |\det(\mathbf{V}')| = 1$, and we are done.

("⇐") For the other direction, assume that there is a unimodular matrix $\mathbf{U}$ such that $\mathbf{B}' = \mathbf{B}\mathbf{U}$. Then, since $\mathbf{U}$ is an integer matrix,

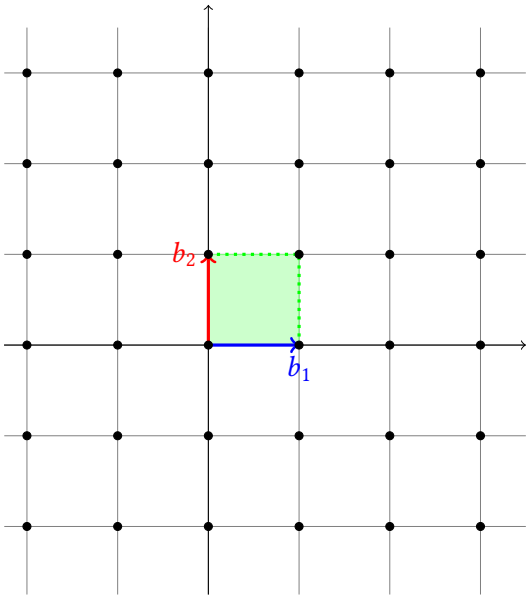
$$\mathcal{L}(\mathbf{B}') \subseteq \mathcal{L}(\mathbf{B})$$

This is because each vector (column) of $\mathbf{B}'$ can be written as a linear combination of vectors in $\mathbf{B}$. Thus, the set of all integer linear combinations of vectors in $\mathbf{B}'$ is contained in the set of all integer linear combinations of vectors in $\mathbf{B}$.

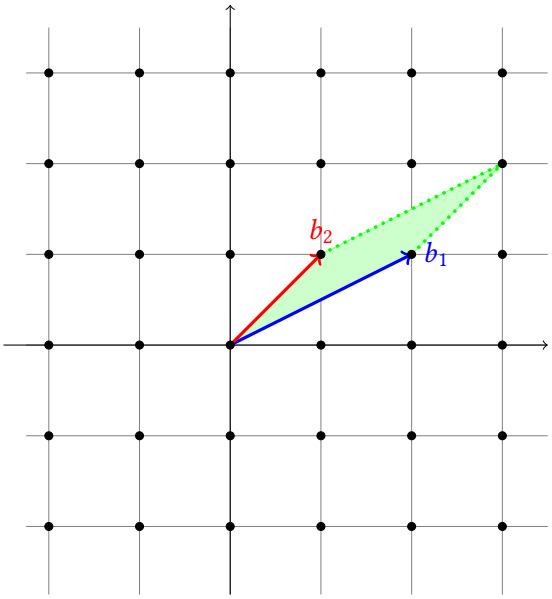
Now, $\mathbf{B} = \mathbf{B}'(\mathbf{U}^{-1})$ where $\mathbf{U}^{-1}$ is also unimodular by Proposition 4. This shows that

$$\mathcal{L}(\mathbf{B}) \subseteq \mathcal{L}(\mathbf{B}')$$

by the same argument as above. Together, we have $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$. □



(a) The lattice $\mathbb{Z}^2$ with basis vectors $(0,1)$ and $(1,0)$ and the associated fundamental parallelepiped.



(b) The lattice $\mathbb{Z}^2$ with a different basis consisting of vectors $(1,1)$ and $(2,1)$, and the associated fundamental parallelepiped.

Figure 2: Parallelepipeds for various bases of the lattice $\mathbb{Z}^2$. Note that the parallelepipeds in either case do not contain any non-zero lattice point.

2.2 A Geometric Characterization using the Fundamental Parallelepiped

We need the notion of a fundamental parallelepiped of a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$.

Definition 6 (Fundamental Parallelepiped)**.** *Given n* linearly independent *vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, their* fundamental parallelepiped *is defined as*

$$\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{R}, 0 \leq x_i < 1 \right\}$$

Thus, pictorially, a fundamental parallelepiped is the (half-open) region enclosed by the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. Clearly, different bases of the same lattice generate different fundamental paralellepipeds. See Figure 2(a) and 2(b).

Note that in Figures 2(a) and 2(b), the vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ form a basis of the lattice, and the parallelepiped associated to the basis does not contain any lattice point other than $\mathbf{0}$. On the other hand, in Figure 3, the vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ do not form a basis of the lattice, and the parallelepiped associated to the basis contains a non-zero lattice point. In fact, this is not a coincidence as our next theorem shows.

Theorem 7. *Let $\mathcal{L}$ be a full-rank n-dimensional lattice, and let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$ denote* linearly independent *vectors in $\mathcal{L}$. Then, $\mathbf{b}_1, \ldots, \mathbf{b}_n$ form a basis of $\mathcal{L}$* if and only if $\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{L} = \{\mathbf{0}\}$.

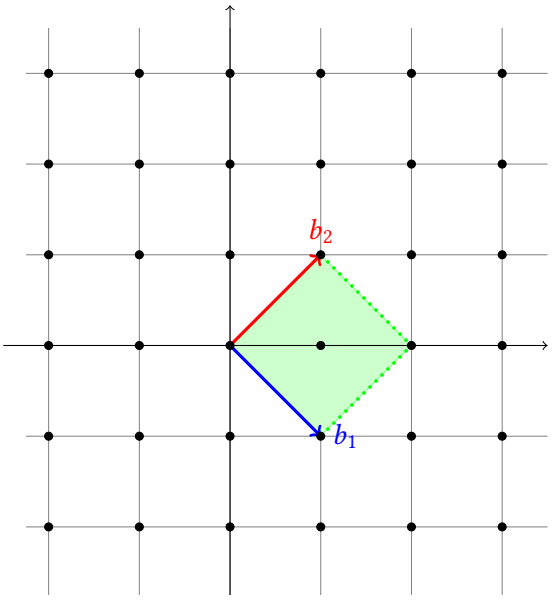


Figure 3: $\mathbf{b}_1$ and $\mathbf{b}_2$ do not form a basis of $\mathbb{Z}^2$. Note that the parallelepiped of $\mathbf{b}_1$ and $\mathbf{b}_2$ contains a non-zero lattice point, namely $(1, 0)$.

Proof. ("⇒") Suppose that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is a basis of $\mathcal{L}$. Let

$$\mathbf{a} = \sum_{i=1}^{n} x_i \mathbf{b}_i \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$$

We will show that $\mathbf{a} = \mathbf{0}$.

Since $\mathbf{a} \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, $x_i \in \mathbb{Z}$ for all i. Since $\mathbf{a} \in \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, $x_i \in [0, 1)$ for all i. Together, this means that $x_i = 0$ for all i, and thus, $\mathbf{a} = \mathbf{0}$.

("⇐") Suppose that $\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{L} = \{\mathbf{0}\}$. We would like to show that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ form a basis of $\mathcal{L}$.

The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent. Since they belong to $\mathcal{L}$, $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \subseteq \mathcal{L}$. What remains is to show that $\mathcal{L} \subseteq \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$. Pick any vector $\mathbf{a} \in \mathcal{L}$ and write it as

$$\mathbf{a} = \sum_{i=1}^{n} x_i \mathbf{b}_i \qquad \text{where } x_i \in \mathbb{R}$$

Consider now the vector

$$\mathbf{a}' = \sum_{i=1}^{n} \lfloor x_i \rfloor \mathbf{b}_i \qquad \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$$

which is clearly in the lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ since the coefficients $\lfloor x_i \rfloor$ are integers. Therefore, the vector $\mathbf{a} - \mathbf{a}'$ is in $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ as well. Now,

$$\mathbf{a} - \mathbf{a}' = \sum_{i=1}^{n} (x_i - \lfloor x_i \rfloor) \mathbf{b}_i \qquad \in \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$$

is in the parallelepiped of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ since $0 \leq x_i - \lfloor x_i \rfloor < 1$ for all i.

Since $\mathbf{a} - \mathbf{a}' \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, it must be the case that $\mathbf{a} - \mathbf{a}' = 0$ by assumption. Since the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent, this means that $x_i - \lfloor x_i \rfloor = 0$ for all i which in turn means that $x_i \in \mathbb{Z}$ for all i.

Thus, $\mathbf{a} \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, showing us that $\mathcal{L} \subseteq \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$. □

2.3 Determinant of a Lattice

Another quantity associated to a lattice is its determinant, denoted $\det(\mathcal{L})$. The determinant of a lattice is the n-dimensional volume of its fundamental parallelepiped, computed as the absolute value of the determinant of its basis matrix $\mathbf{B}$. A couple of facts about the determinant of a lattice are worth noting:

1. The parallelepipeds associated with different bases of a lattice have the same volume. Thus, *the determinant is a lattice invariant.* This is easy to see using our characterization of equivalent bases from Theorem 5.

   Let $\mathbf{B}$ and $\mathbf{B}'$ be any two lattice bases. By Theorem 5, there is a unimodular matrix $\mathbf{U}$ such that $\mathbf{B}' = \mathbf{BU}$. Thus, $|\det(\mathbf{B}')| = |\det(\mathbf{B})| \cdot |\det(\mathbf{U})| = |\det(\mathbf{B})|$ since $|\det(\mathbf{U})| = 1$.

2. Intuitively, the determinant of a lattice is inversely proportional to its "density". The larger the determinant, the sparser the lattice.

3 Gram-Schmidt Orthogonalization

Gram-Schmidt orthogonalization is a procedure in linear algebra that transforms a set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ into a set of orthogonal vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$. In two dimensions, this proceeds as follows:

- The first Gram-Schmidt vector $\tilde{\mathbf{b}}_1$ is $\mathbf{b}_1$ itself.
- The second Gram-Schmidt vector $\tilde{\mathbf{b}}_2$ is the component of $\mathbf{b}_2$ that is orthogonal to $\mathsf{Span}(\tilde{\mathbf{b}}_1)$. This can be computed as

  $$\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - \left(\frac{\langle \mathbf{b}_2, \tilde{\mathbf{b}}_1 \rangle}{\langle \tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_1 \rangle} \right) \tilde{\mathbf{b}}_1$$

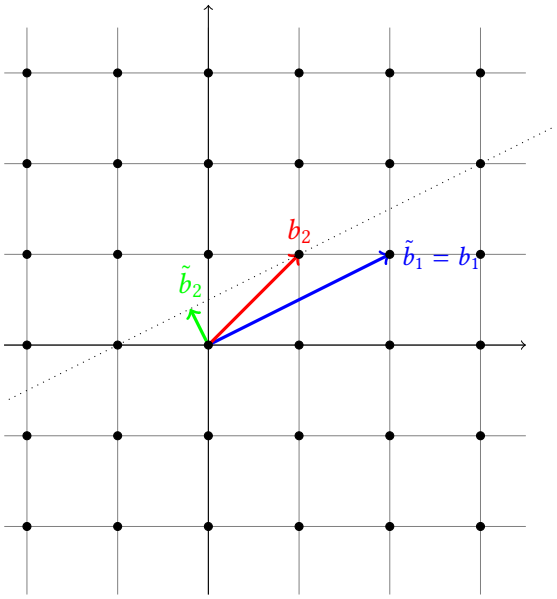
  See Figure 4 for an illustration of this process.

In general, the Gram-Schmidt vectors are obtained by projecting each vector successively on the space orthogonal to the span of all the previous vectors.

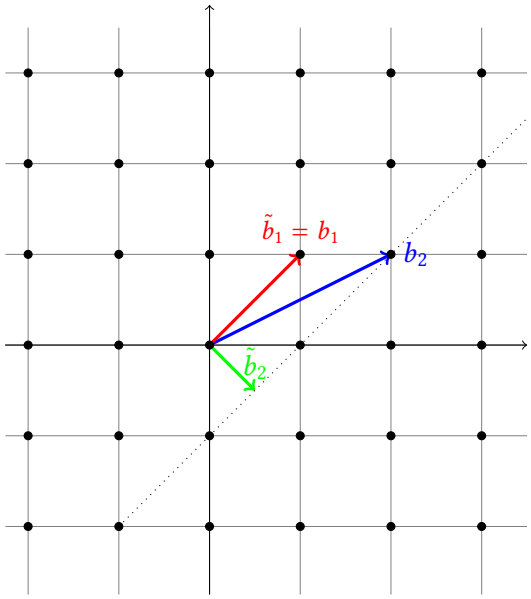
Definition 8 (Gram-Schmidt Orthogonalization)**.** *For a sequence of n linearly independent vectors* $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$*, we define their* Gram-Schmidt orthogonalization *as the sequence of vectors* $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ *defined as follows:*

$$\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \tilde{\mathbf{b}}_j \qquad \text{where } \mu_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$$

Thus, $\tilde{\mathbf{b}}_j$ *is the component of* $\mathbf{b}_i$ *that is orthogonal to* $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{i-1}$*. The coefficients* $\mu_{i,j}$ *are called the Gram-Schmidt coefficients.*



(a) Gram-Schmidt orthogonalization of the vectors b_1 and b_2 in that order.



(b) Gram-Schmidt orthogonalization of the same vectors, but in the opposite order.

Figure 4: Gram-Schmidt Orthogonalization.

Remarks.

1. True to its name, the different Gram-Schmidt vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ are orthogonal to each other. That is, for each $i \neq j$, $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0$. This is an easy consequence of Definition 8.
2. The span of $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_i$ is the same as the span of $\mathbf{b}_1, \ldots, \mathbf{b}_i$ for all $1 \leq i \leq n$.
3. The vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ do not form a lattice basis. In fact, the Gram-Schmidt vectors are not necessarily in the lattice. See Figure 4 for example.
4. The (Euclidean) length of the Gram-Schmidt vector $\tilde{\mathbf{b}}_i$ is at most the length of the basis vector $\mathbf{b}_i$. Namely, $\|\tilde{\mathbf{b}}_i\| \leq \|\mathbf{b}_i\|$.
5. Clearly, as seen in Figure 4, the Gram-Schmidt vectors depend on the order in which the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are processed.

Let $\tilde{\mathbf{b}}_1/\|\tilde{\mathbf{b}}_1\|, \ldots, \tilde{\mathbf{b}}_n/\|\tilde{\mathbf{b}}_n\|$ denote the unit vectors in the direction of the Gram-Schmidt vectors. Then, the

Gram-Schmidt orthogonalization process can be written in matrix form as

$$\begin{pmatrix} | & & | \\ \mathbf{b}_1 & \dots & \mathbf{b}_n \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ \tilde{\mathbf{b}}_1 & \dots & \tilde{\mathbf{b}}_n \\ | & & | \end{pmatrix} \cdot \begin{pmatrix} 1 & \mu_{2,1} & \mu_{3,1} & \dots & \mu_{n,1} \\ 0 & 1 & \mu_{3,2} & \dots & \mu_{n,2} \\ 0 & 0 & 1 & \dots & \mu_{n,3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} | & & | \\ \frac{\tilde{\mathbf{b}}_1}{\|\tilde{\mathbf{b}}_1\|} & \dots & \frac{\tilde{\mathbf{b}}_n}{\|\tilde{\mathbf{b}}_n\|} \\ | & & | \end{pmatrix} \cdot \begin{pmatrix} \|\tilde{\mathbf{b}}_1\| & \mu_{2,1}\|\tilde{\mathbf{b}}_1\| & \mu_{3,1}\|\tilde{\mathbf{b}}_1\| & \dots & \mu_{n,1}\|\tilde{\mathbf{b}}_1\| \\ 0 & \|\tilde{\mathbf{b}}_2\| & \mu_{3,2}\|\tilde{\mathbf{b}}_2\| & \dots & \mu_{n,2}\|\tilde{\mathbf{b}}_2\| \\ 0 & 0 & \|\tilde{\mathbf{b}}_3\| & \dots & \mu_{n,3}\|\tilde{\mathbf{b}}_3\| \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \|\tilde{\mathbf{b}}_n\| \end{pmatrix}$$

Since the vectors $\frac{\tilde{\mathbf{b}}_i}{\|\tilde{\mathbf{b}}_i\|}$ are orthonormal, the determinant of the matrix with columns $\frac{\tilde{\mathbf{b}}_i}{\|\tilde{\mathbf{b}}_i\|}$ is 1.

Thus, we have

$$\det(\mathcal{L}(\mathbf{B})) = \prod_{i=1}^{n} \|\tilde{\mathbf{b}}_i\|$$

In other words, the Gram-Schmidt orthogonalization process is a volume-preserving transformation that results in a set of orthogonal vectors $\tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_n$, whose enclosing parallelepiped is rectangular and generates a volume of $\det(\mathcal{L}(\mathbf{B}))$.

4 Successive Minima of a Lattice

A basic parameter of the lattice is the length of the shortest non-zero vector in the lattice (since any lattice contains the zero vector which has norm zero, we have to ask for a non-zero vector). This parameter is also called the *first successive minimum* of the lattice, and is denoted $\lambda_1(\mathcal{L})$. When we speak of length, we mean the Euclidean norm defined as follows: for a vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, the Euclidean norm of $\mathbf{x}$, denoted $\|\mathbf{x}\|_2$ (or simply as $\|\mathbf{x}\|$ is defined as

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

The Euclidean norm is also frequently referred to as the ℓ_2 norm. We can speak of other norms such as the ℓ_1 norm – $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$ – and the ℓ_∞ norm – $\|\mathbf{x}\|_\infty = \max_{i=1}^{n} |x_i|$, but we will stick to the Euclidean norm for most of this course.

Figure 5 shows a shortest vector in the lattice generated by $(1, 1)$ and $(2, 0)$. The shortest vector is not unique in general. There could be many, even exponentially many, shortest vectors. Clearly, there are at least two – if $\mathbf{v}$ is a shortest vector in a lattice, then so is $-\mathbf{v}$.

We will be interested in lower and upper bounds on λ_1. We first show a lower bound on λ_1 using Gram-Schmidt orthogonalization. Then, we will prove Minkowski's theorem which provides an upper bound on λ_1 in terms of the determinant of the lattice.

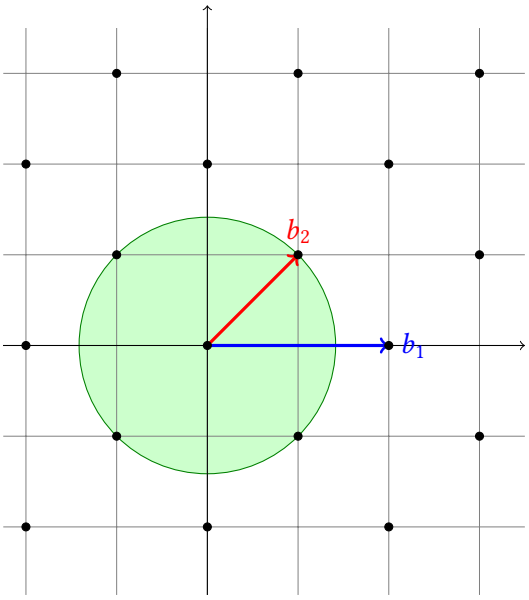


Figure 5: The shortest vector in the lattice generated by $(1, 1)$ and $(2, 0)$. $\lambda_1(\mathcal{L}) = \sqrt{2}$.

Lower Bound on λ_1. We show that the shortest non-zero vector in a lattice is at least as long as the shortest Gram-Schmidt vector of (any) basis of the lattice. To see why, observe that a lattice can be partitioned into many hyperplanes perpendicular to its Gram-Schmidt vector $\tilde{\mathbf{b}}_n$. See Figure 6 for an illustration in two dimensions. Now, there are two possibilities:

- There is a shortest non-zero vector in one of the hyper-planes not passing through the origin. In that case, the vector has to have length at least $\|\tilde{\mathbf{b}}_n\| \geq \min_j \|\tilde{\mathbf{b}}_j\|$ since the i^{th} such hyper-plane is at a distance of $i \cdot \|\tilde{\mathbf{b}}_n\|$ from the origin.
- The shortest non-zero vector lives in the hyper-plane that passes through the origin, in which case, repeat the same argument in dimension $n - 1$ with the $(n - 1)$-dimensional sublattice partitioned into hyper-planes perpendicular to $\tilde{\mathbf{b}}_{n-1}$.

Eventually, if the argument reaches dimension 1, the shortest non-zero vector has to have length at least $\|\mathbf{b}_1\| = \|\tilde{\mathbf{b}}_1\| \geq \min_j \|\tilde{\mathbf{b}}_j\|$.

The formal statement and proof of the theorem follows.

Theorem 9. *Let* $\mathbf{B}$ *be a rank-n lattice basis, and* $\tilde{\mathbf{B}}$ *be its Gram-Schmidt orthogonalization. Then,*

$$\lambda_1(\mathcal{L}(\mathbf{B})) \geq \min_{i=1,\ldots,n} \|\tilde{\mathbf{b}}_i\| > 0$$

Proof. Let $\mathbf{x} \in \mathbb{Z}^n$ be *any* non-zero integer vector. We would like to show that the lattice vector $\mathbf{Bx} \in \mathcal{L}(\mathbf{B})$ has length at least $\min_i \|\tilde{\mathbf{b}}_i\|$.

The proof follows by calculating the quantity $|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j \rangle|$ in two different ways.

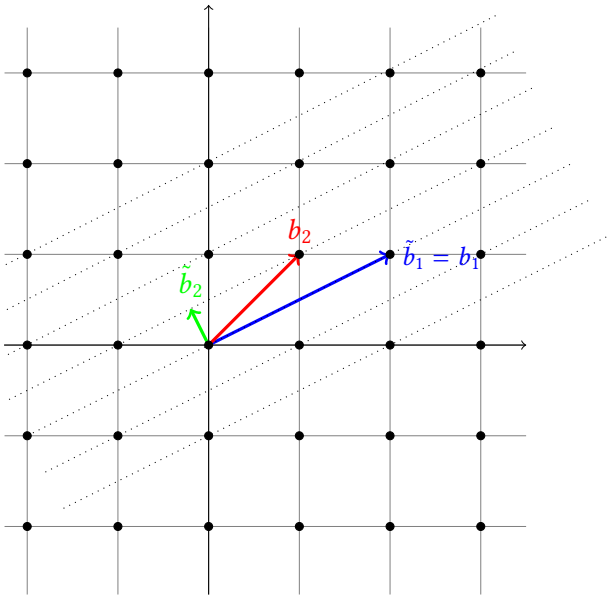


Figure 6: The lattice is partitioned into many parallel hyperplanes perpendicular to $\tilde{\mathbf{b}}_2$. Either the shortest vector lives in a hyperplane that does not pass through the origin, in which case its length is at least $\|\tilde{\mathbf{b}}_2\|$ or it lives in the hyperplane that passes through the origin, in which case its length is at least $\tilde{\mathbf{b}}_1 = \|\tilde{\mathbf{b}}_1\|$. In general, in two dimensions, $\lambda_1(\mathcal{L}) \geq \min\{\|\tilde{\mathbf{b}}_1\|, \|\tilde{\mathbf{b}}_2\|\}$. This argument can be generalized to n dimensions.

1. Let $j \in \{1, \ldots, n\}$ be the largest index such that $x_j \neq 0$. Then,

$$|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j \rangle| = |\langle \sum_{i=1}^{n} x_i \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle| = |\sum_{i=1}^{n} x_i \langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle| = |x_j| \langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle = |x_j| \cdot \|\tilde{\mathbf{b}}_j\|^2 \tag{2}$$

where the first equality follows by rewriting $\mathbf{Bx}$ as $\sum_{i=1}^{n} x_i \mathbf{b}_i$, the second follows by the linearity of the inner product, and the third because

- for $j < i$, $\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle = 0$
- for $j > i$, $x_j = 0$ by the definition of j.

The fourth equality follows by the definition of $\|\tilde{\mathbf{b}}_j\|^2 = \langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle$.

2. On the other hand,

$$|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j \rangle| \leq \|\mathbf{Bx}\| \cdot \|\tilde{\mathbf{b}}_j\| \tag{3}$$

by the Cauchy-Schwarz inequality.

Putting together Equations 2 and 3, we get

$$\|\mathbf{Bx}\| \geq \frac{|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j \rangle|}{\|\tilde{\mathbf{b}}_j\|} = |x_j| \cdot \|\tilde{\mathbf{b}}_j\| \geq \|\tilde{\mathbf{b}}_j\| \geq \min_{i=1\ldots n} \|\tilde{\mathbf{b}}_i\|$$

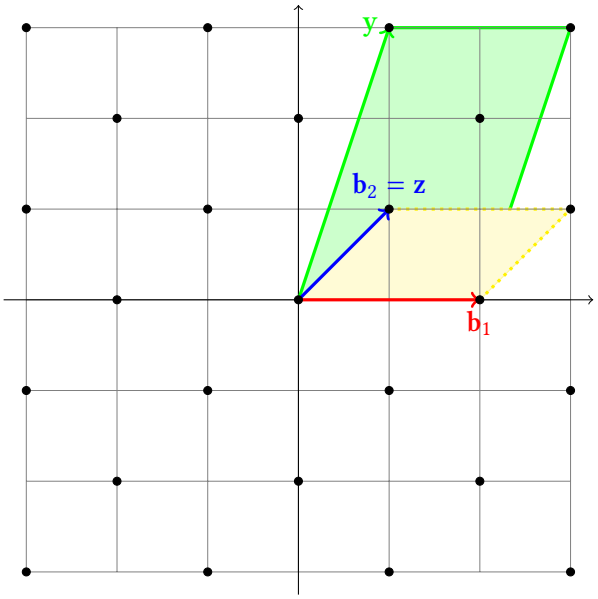


Figure 7: Constructing lattice basis from a discrete additive subgroup of . After the first iteration if we choose $\mathbf{y} = (1, 3)$, then in the $\mathcal{P}(\mathbf{b}_1, \mathbf{y})$ we choose $\mathbf{z} = \mathbf{b}_2$ which is at the minimum distance from $\mathsf{Span}(\mathbf{b}_1)$. We can see that there are no non-zero lattice vectors in $\mathcal{P}(\mathbf{b}_1, \mathbf{b}_2)$. Therefore, $\{\mathbf{b}_1, \mathbf{b}_2\}$ forms a basis for this lattice.

where the third inequality follows from the fact that x_j is a non-zero integer. Since the length of any lattice vector is at least $\min_i \|\tilde{\mathbf{b}}_i\|$,

$$\lambda_1(\mathbf{B}) \geq \min_{i=1\ldots n} \|\tilde{\mathbf{b}}_i\|$$

Since $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent, this quantity is strictly positive. □

A corollary of this theorem is that a lattice is a *discrete set*. In other words, lattice points cannot be arbitrarily close to one another. Formally:

Corollary 10. *For every lattice $\mathcal{L}$, there is an $\varepsilon = \varepsilon(\mathcal{L}) > 0$ such that $\|\mathbf{x} - \mathbf{y}\| \geq \varepsilon$ for any two unequal lattice points $\mathbf{x}, \mathbf{y} \in \mathcal{L}$.*

Proof. For any two $\mathbf{x} \neq \mathbf{y} \in \mathcal{L}$, $\mathbf{x} - \mathbf{y} \in \mathcal{L}$. Then, $\|\mathbf{x} - \mathbf{y}\| \geq \lambda_1(\mathcal{L}) > 0$. In particular, set $\varepsilon = \lambda_1(\mathcal{L})$ to obtain the statement of the corollary. □

A Basis-Independent Definition. In fact, this leads us to a *basis-independent* characterization of a lattice. Namely, every discrete subset of $\mathbb{R}^n$ that is closed under subtraction is a lattice. We will omit the proof and refer to the Micciancio-Goldwasser book for details.

5 Minkowski's Theorems

Before we state Minkowski's theorem, we give some intuition on what the length of a shortest lattice vector might depend on. It stands to reason that denser lattices must have shorter vectors.

Recall that the determinant of a lattice is inversely proportional to its density: a larger determinant implies a less dense lattice. Therefore, we should be able to express an upper bound for λ_1 in terms of the determinant of a lattice.

It is not hard to see that $\lambda_1 \leq \det(\mathcal{L})$ (Do you see why?). However, this cannot be the right bound; for example, it does not scale correctly. In particular, if $\mathcal{L}$ is an arbitrary lattice with a shortest vector of length λ_1, then lattice defined by scaling every vector in $\mathcal{L}$ by k should have a shortest vector of length $k\lambda_1$. However, $det(k\mathcal{L}) = k^n\mathcal{L}$.

We will now prove a much stronger bound, namely Minkowski's first theorem. It is not hard to check that Minkowski's first theorem scales correctly.

Theorem 11 (Minkowski's First Theorem)**.** *For every full rank lattice $\mathcal{L}$,*

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

In order to prove the theorem, we will need to use the following two lemmas.

Lemma 12 (Blichfeld)**.** *For all full rank lattice $\mathcal{L}$ and measurable set $S \subseteq \mathbb{R}^n$ s.t.* $\mathrm{vol}(S) > \det(\mathcal{L})$*,*

$$\exists\, \mathbf{x}, \mathbf{y} \in S, \textit{s.t. } \mathbf{x} - \mathbf{y} \in \mathcal{L}.$$

See Figure-8 for an example.

Proof. Let $\mathbf{B}$ be a basis for the lattice $\mathcal{L}$. Define $f : \mathbb{R}^n \to \mathcal{P}(\mathbf{B})$ as follows:

$$f(\sum x_i \mathbf{b}_i) = \sum (x_i - \lfloor x_i \rfloor)\mathbf{b}_i \ .$$

First, note that $\sum x_i\mathbf{b}_i - f(\sum x_i\mathbf{b}_i) = \sum \lfloor x_i \rfloor \mathbf{b}_i \in \mathcal{L}$. Now consider the following 2 cases:

- Case 1: If $\exists \mathbf{x}, \mathbf{y} \in S$ s.t. $f(\mathbf{x}) = f(\mathbf{y})$ (i.e. we have a collision from two vectors), then, $\mathbf{x} - \mathbf{y} = (\mathbf{x} - f(\mathbf{x})) - (\mathbf{y} - f(\mathbf{y}))$. But as noted above, $\mathbf{x} - f(\mathbf{x}) \in \mathcal{L}$ and $\mathbf{y} - f(\mathbf{y}) \in \mathcal{L}$. Therefore, $\mathbf{x} - \mathbf{y} \in \mathcal{L}$.

- Case 2: Assume there are no collisions. Let $S = \bigcup_{\mathbf{x} \in \mathcal{L}} S_{\mathbf{x}}$. Define $\tilde{S_{\mathbf{x}}} = S_{\mathbf{x}} - \mathbf{x}$. By definition, $\tilde{S_{\mathbf{x}}} \subseteq \mathcal{P}(\mathbf{B})$. Also,
$$\mathrm{vol}(S) = \sum_{\mathbf{x} \in \mathcal{L}} \mathrm{vol}(S_{\mathbf{x}}) \text{ and } \mathrm{vol}(\tilde{S_{\mathbf{x}}}) = \mathrm{vol}(S_{\mathbf{x}}) \ .$$
Therefore, $\mathrm{vol}(S) = \sum \mathrm{vol}(S_{\mathbf{x}}) = \sum \mathrm{vol}(\tilde{S_{\mathbf{x}}})$. But since we assume that we do not have any collisions, for all $\mathbf{x}, \mathbf{y}$, $\tilde{S_{\mathbf{x}}} \bigcap \tilde{S_{\mathbf{y}}} = \emptyset$. And so,
$$\mathrm{vol}(S) = \sum \mathrm{vol}(\tilde{S_{\mathbf{x}}}) = \mathrm{vol}(\bigcup_{\mathbf{x} \in \mathcal{L}} \tilde{S_{\mathbf{x}}}) \leq \mathrm{vol}(\mathcal{P}(B)) = \det(\mathcal{L})$$
Therefore, $\mathrm{vol}(S) \leq \det(\mathcal{L})$ which contradicts with our assumption.

□

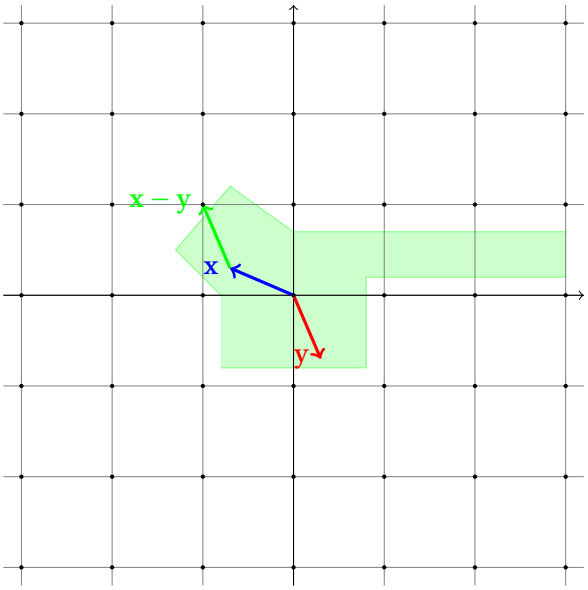


Figure 8: By the Blichfeld's theorem we can find $\mathbf{x}$ and $\mathbf{y}$ in this set such that $\mathbf{x} - \mathbf{y} \in \mathcal{L}$.

Definition 13 (Convex Set)**.** *A set S is convex if:*

$$\forall x \neq y \in S, \forall \alpha \in [0,1], \alpha x + (1-\alpha)y \in S.$$

That is, if we take any two points from a convex set, any point that lies on the straight line between the two points must also be in the set.

Definition 14 (Centrally Symmetric Set)**.** *A set S is centrally symmetric if:*

$$\forall x \in S, -x \in S.$$

Theorem 15 (Minkowski's Convex Body Theorem)**.** *For all full-rank lattice $\mathcal{L}$, and a convex centrally symmetric set S with* $\mathrm{vol}(S) > 2^n \det(\mathcal{L})$*, S contains a non-zero lattice point.*

Proof. Let $\tilde{S} = \{\mathbf{x}/2 : \mathbf{x} \in S\}$. Then,

$$\mathrm{vol}(\tilde{S}) = 2^{-n} \cdot \mathrm{vol}(S) > \det(\mathcal{L})$$

Therefore, by the Blichfeld's theorem $\exists \mathbf{x}, \mathbf{y} \in \tilde{S}$ s.t. $\mathbf{x} - \mathbf{y} \in \mathcal{L}$. We will show that $\mathbf{x} - \mathbf{y} \in S$. Now, $2\mathbf{x} \in S$ and $2\mathbf{y} \in S$ by the construction of $\tilde{S}$. Therefore, $-2\mathbf{y} \in S$ by central symmetry, and $\mathbf{x} - \mathbf{y} = \frac{2\mathbf{x} - 2\mathbf{y}}{2} \in S$ by convexity of S.

□

We are now ready to prove Minkowski's first theorem (Theorem 1).

Proof. Let $S = \mathcal{B}(0, \lambda_1)$, where $\mathcal{B}(\mathbf{x}, r)$ is an n-dimensional open ball of radius r centred at $\mathbf{x}$. Note that

$$\mathrm{vol}(\mathcal{B}(0,r)) \geq \left(\frac{2r}{\sqrt{n}}\right)^n.$$

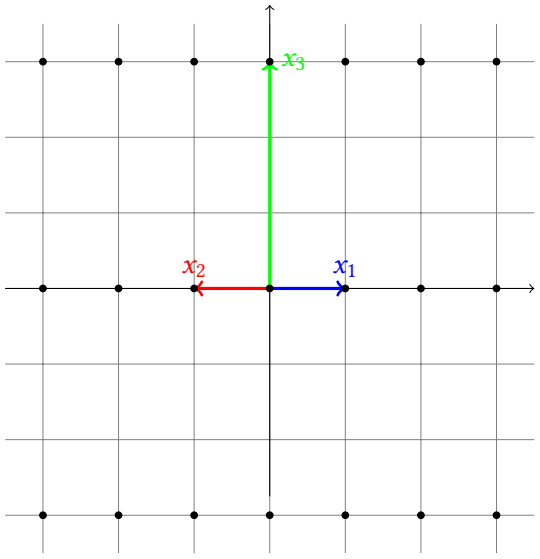


Figure 9: The first and second successive minima in the lattice generated by $(1, 0)$ and $(0, 3)$. Knowing that $\lambda_1 = \|\mathbf{x}_1\|$, we can ask whether $\lambda_2 = \|\mathbf{x}_2\|$, $\lambda_2 = 2\lambda_1$ or $\lambda_2 = \|\mathbf{x}_3\|$. By the definition, $\lambda_2 = \|\mathbf{x}_3\|$.

since this n-dimensional ball contains an n-dimensional cube of length $\frac{2r}{\sqrt{n}}$. From Minkowski's convex body theorem and the fact that the ball is open and hence contains no non-zero lattice points, we get

$$\left(\frac{2\lambda_1}{\sqrt{n}}\right)^n \leq \text{vol}(\mathcal{B}(0, \lambda_1)) \leq 2^n \det(\mathcal{L}).$$

Rearranging,

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

□

Successive Minima. Given a lattice, it is natural to ask for a second shortest vector in the lattice and in general, the ith shortest vector in the lattice?

Definition 16 (Successive Minima)**.** *Let $\mathcal{L}$ be an arbitrary lattice of rank n. Then $\forall i, 1 \leq i \leq n$:*

$$\lambda_i \stackrel{\text{def}}{=} \inf\{r \,:\, \mathcal{B}(0, r) \text{ contains } \geq i \text{ linearly independent lattice vectors}\}$$

Following the above the definition and the lattice described in Figure-9, we can see that $\lambda_1 = \|\mathbf{x}_1\|$ and $\lambda_2 = \|\mathbf{x}_3\|$, since neither $\mathbf{x}_2$ nor $2\mathbf{x}_1$ are linearly independent of $\mathbf{x}_1$.

We saw that Minkowski's first theorem gives us an upper bound on λ_1. In fact, his second theorem strengthens the results by considering a geometric mean of $\lambda_1, \lambda_2, \ldots, \lambda_n$. For a proof, we refer the reader to Oded Regev's lecture notes.

Theorem 17 (Minkowski's Second Theorem)**.** *For all full rank lattices $\mathcal{L}$,*

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

6 Computational Problems

Now that we are comfortable with basic mathematical theorems related to lattices, we will move on to defining computational problems on lattices. In particular, we will consider several variants of the Shortest Vector Problem (SVP), the Closest Vector Problem (CVP) and the Shortest Independent Vectors Problem (SIVP) and show relations among them.

6.1 "Hard" Lattice Problems

Recall that $\lambda_i = \lambda_i(\mathcal{L}(\mathbf{B}))$ denotes the i-th successive minimum of the lattice $\mathcal{L}(\mathbf{B})$ and that $\mathsf{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) := \min_{\mathbf{u}\in\mathcal{L}(\mathbf{B})} \|\mathbf{u} - \mathbf{t}\|$. We will always talk about full rank lattices and the Euclidean (ℓ_2) norm, unless otherwise noted. Most of our discussion generalizes readily to the more general cases.

Shortest Vector Problem (SVP) The shortest vector problem (SVP) is simply to find the shortest non-zero vector in a lattice $\mathcal{L}(\mathbf{B})$ given the basis $\mathbf{B} \in \mathbb{Z}^{n\times n}$. More precisely, given $\mathbf{B} \in \mathbb{Z}^{n\times n}$, one has to find a vector $\mathbf{u} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{u}\| = \lambda_1$. We will also define the (successively easier) optimization and decision versions of SVP. The optimization version asks to find $\lambda_1(\mathcal{L}(\mathbf{B}))$ given $\mathbf{B}$. In the decision version, one is given $\mathbf{B}$ and a number $d \in \mathbb{R}$ and is asked to decide if $\lambda_1(\mathcal{L}(\mathbf{B})) \le d$.

Of great interest to us are the approximate versions of these problems. In words, the γ-approximate shortest vector problem SVP_γ asks to find a vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}\| \le \gamma \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$. Its decisional version gapSVP_γ (for $\gamma \ge 1$) asks to distinguish whether a given lattice basis $\mathbf{B}$ generates a lattice $\mathcal{L}(\mathbf{B})$ with $\lambda_1(\mathcal{L}) \le 1$ or $\lambda_1(\mathcal{L}) > \gamma$.

Closest Vector Problem (CVP) The closest vector problem (CVP) is to find the closest vector in a lattice $\mathcal{L}(\mathbf{B})$ to a given target point $\mathbf{t} \in \mathbb{R}^n$, given the basis $\mathbf{B} \in \mathbb{Z}^{n\times n}$ and the target $\mathbf{t}$.

Shortest Independent Vectors Problems (SIVP) given a lattice basis $\mathbf{B}$, find n independent and "short" vectors. That is, find vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ where $\|\mathbf{v}_i\| \le \lambda_n$. Note that the problem does not ask for vectors of length $\lambda_1, \dots, \lambda_n$.

6.2 Complexity Landscape

We present the landscape for the Shortest Vector Problem. The landscape for CVP and SIVP are very similar. In the following we assume we are given a lattice basis in $\mathbb{Z}^n$. The runtimes of the described algorithms will be a function of n, and we ignore polynomial factors in the length of the bit representation of the given basis.

Algorithms for SVP_γ: The first algorithm to solve the SVP_γ was the LLL algorithm. It gave a $2^{O(n)}$-approximation and its running time is poly(n). The best known approximation factor achieved by a polynomial time algorithm is $2^{O\left(\frac{n \log\log n}{\log n}\right)}$.

If we want to solve the exact SVP (i.e., $\gamma = 1$), then the LLL algorithm can do so with running time of $2^{O(n^2)}$. The fastest known algorithm to solve the exact SVP was given by Aggarwal, Dadush, Regev and Stephens-Davidowitz in 2015, and its running time is $2^{n+o(n)}$. There are also heuristic algorithms with runtime $2^{0.292n}$.

Finally, one can have a trade off between the approximation factor and the running time — achieving 2^k-approximation can be done in $2^{\tilde{O}(n/k)}$ time.

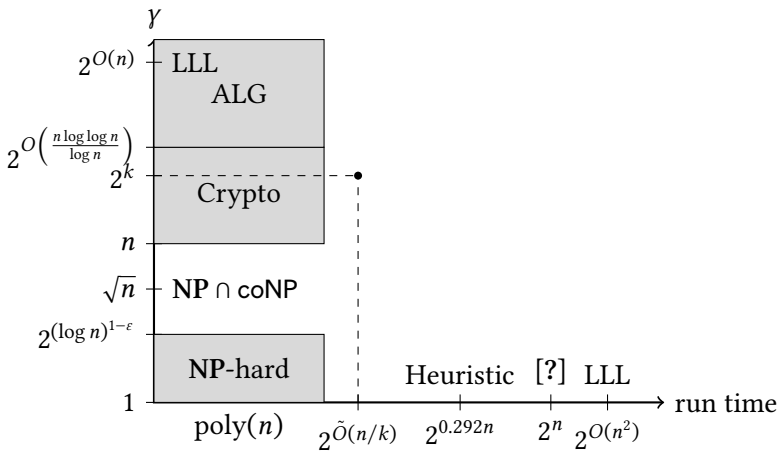


Figure 10: The complexity landscape of SVP_γ.

Hardness of SVP_γ: It is no surprise that we don't know of a polynomial time algorithm to solve the exact SVP, since it was shown to be **NP**-hard. In fact, even achieving $2^{(\log n)^{1-\varepsilon}}$-approximation is **NP**-hard. On the other hand, it was shown that $\mathsf{SVP}_{\sqrt{n}}$ is in **NP** ∩ coNP, and thus unlikely to be **NP**-hard.

Cryptography from SVP_γ: The smallest approximation factor from which we know how to build cryptographic primitives is $\gamma = n$. Since $\mathsf{SVP}_{\sqrt{n}} \in$ **NP**∩coNP, this will likely not suffice to base cryptography on **NP**-hardness.

A pictorial presentation of the above description if given in Figuire 10.

7 The LLL Algorithm

We describe the classic Lenstra-Lenstra-Lovász (henceforth called LLL) algorithm, a polynomial-time algorithm for computing a $2^{O(n)}$-approximation of the shortest lattice vector. We will then show several applications of the LLL algorithm such as computing the shortest vector exactly in time $2^{O(n^2)}$; Babai's algorithms that use LLL to come up with a $2^{O(n)}$-approximation for the closest vector problem; an outline of Lenstra's integer programming algorithm; as well as applications in cryptanalysis.

7.1 The Algorithm

For a basis $\mathbf{B}$, let $W_{\mathbf{B}}$ denote the length of the bit representation of $\mathbf{B}$.

Theorem 18. *Given a basis $\mathbf{B} \in \mathbb{Z}^{n \times n}$ there is a $\mathrm{poly}(n, W_{\mathbf{B}})$-time algorithm for $\mathsf{SVP}_{2^{O(n)}}$, where $W_{\mathbf{B}}$ is the length of the bit representation of $\mathbf{B}$; namely, the algorithm returns a vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}\| \leq 2^{O(n)} \cdot \lambda_1(\mathbf{B})$.*

The LLL algorithm actually transforms, in polynomial time, the given basis into a "LLL-reduced" basis for the same lattice. The above theorem holds since an LLL-reduced basis has an important property, namely, its shortest vector is a $2^{O(n)}$-approximation for the shortest vector in the entire lattice. We first define an LLL-reduced basis and give some intuition for this definition. Subsequently, we will describe and analyze the LLL algorithm which finds an LLL-reduced basis in time polynomial in the input size.

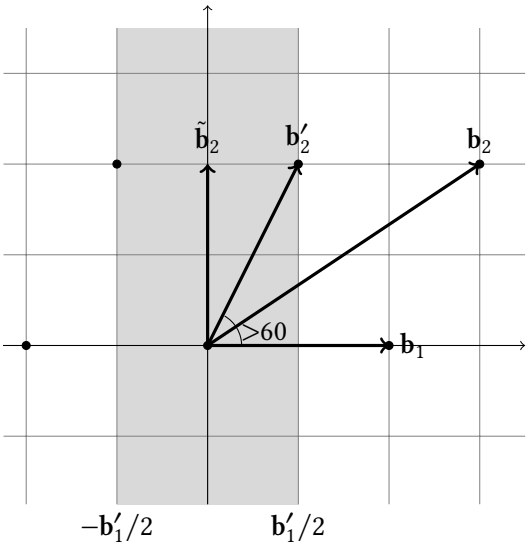


Figure 11: The LLL-reduced basis of $[\mathbf{b}_1 = (2,0), \mathbf{b}_2 = (3,2)]$ is $[\mathbf{b}'_1 = (2,0), \mathbf{b}'_2 = (1,2)]$.

7.2 LLL-reduced Basis

Our goal is to transform the given basis to one with "short" vectors. Consider the case $n = 2$, i.e., $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$. Our starting point is the Gram-Schmidt orthogonalization process in which we set $\tilde{\mathbf{b}}_1 := \mathbf{b}_1$ and $\tilde{\mathbf{b}}_2 := \mathbf{b}_2 - \mu_{2,1}\tilde{\mathbf{b}}_1$, where $\mu_{2,1} = \langle \mathbf{b}_2, \tilde{\mathbf{b}}_1 \rangle / \langle \tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_1 \rangle$.

Intuitively, $\tilde{\mathbf{b}}_2$ is the shortest vector we can hope for (outside the span of $\mathbf{b}_1$) since we removed all $\tilde{\mathbf{b}}_1$'s components from it; this fact is what makes the Gram-Schmidt orthogonal. However, $\tilde{\mathbf{b}}_2 \notin \mathcal{L}(\mathbf{B})$, and thus cannot be in a basis of $\mathcal{L}(\mathbf{B})$. To fix this issue, we transform $\mathbf{B}$ into the following basis: $\mathbf{b}'_1 = \mathbf{b}_1$ and $\mathbf{b}'_2 = \mathbf{b}_2 - \lfloor \mu_{2,1} \rceil \mathbf{b}'_1$, where $\lfloor \cdot \rceil$ means rounding to the closest integer. $\mathbf{b}'_2$ is the shortest *lattice* vector we can hope for, as we removed all the *integer* components of $\mathbf{b}_1$. Note that the projection of $\mathbf{b}'_2$ to the line generated by $\mathbf{b}'_1$ is between $-\mathbf{b}'_1/2$ to $\mathbf{b}'_1/2$. See Figure 11 for an example of this transformation.

So far we reduced $\mathbf{b}_2$, but left $\mathbf{b}_1$ as is. But what if $\mathbf{b}_1$ was very long to begin with? There is no guarantee that the reduced basis is short. At this point we adopt an idea from Euclid's greatest common divisor (gcd) algorithm: we reduce $\mathbf{b}_2$ with respect to $\mathbf{b}_1$ as much as we can, then swap the roles of $\mathbf{b}_1$ and $\mathbf{b}_2$ and repeat the process. The process will stop when the basis meets the following conditions, which form the definition of an LLL-reduced basis in two dimensions.

Definition 19 (LLL-reduced basis in two dimensions)**.** *A basis* $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$ *is LLL-reduced if*

1. $|\mu_{2,1}| \leq 1/2$.
2. $\|\mathbf{b}_2\| \geq \|\mathbf{b}_1\|$.

Note that the second condition can be written as

$$\left\|\tilde{\mathbf{b}}_2\right\|^2 \geq (1 - \mu_{21}^2) \left\|\tilde{\mathbf{b}}_1\right\|^2 .$$

This condition also guarantees that the resulting basis is "close" to orthogonal, in particular, that the angle between $\mathbf{b}_1$ and $\mathbf{b}_2$ is at least 60 degrees.

Generalizing this to n dimensions, we get the following definition.

Definition 20 (δ-LLL-reduced basis)**.** *Let $\delta \in (1/4, 1)$. A basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is δ-LLL-reduced if*

1. ***size-reduced.*** *$|\mu_{i,j}| \leq 1/2$ for every $i > j$.*
2. ***Lovász criterion.*** *$\left\|\tilde{\mathbf{b}}_{i+1}\right\|^2 \geq (\delta - \mu_{i+i,i}^2)\left\|\tilde{\mathbf{b}}_i\right\|^2$ for every $1 \leq i \leq n-1$.*

Note that the projection of a (partial) LLL-reduced basis $[\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_i, \mathbf{b}_{i+1}]$ to $\mathsf{Span}(\tilde{\mathbf{b}}_i, \ldots, \tilde{\mathbf{b}}_n)$ is

$$[0, \ldots, 0, \tilde{\mathbf{b}}_i, \mathbf{b}_{i+1} + \mu_{i+1,i}\tilde{\mathbf{b}}_i] \,.$$

The last two vectors meet the definition of LLL-reduced basis in two dimensions.

To get some intuition for this definition, we look at the 2D variant above where we said that $\|\mathbf{b}_2\| \geq \|\mathbf{b}_1\|$ or equivalently

$$\left\|\tilde{\mathbf{b}}_2\right\|^2 \geq (1 - \mu_{2,1}^2)\left\|\tilde{\mathbf{b}}_1\right\|^2 \,.$$

We are slightly changing this by adding δ as a parameter. Geometrically, the criterion looks at the projection of vectors $\mathbf{b}_1, \mathbf{b}_2 \cdots \mathbf{b}_n$ on to the Gram-Schmidt vectors $\tilde{\mathbf{b}}_i, \ldots, \tilde{\mathbf{b}}_n$. The first few vectors $\mathbf{b}_1 \cdots \mathbf{b}_{i-1}$ project to 0, $\mathbf{b}_i$ becomes $\tilde{\mathbf{b}}_i$ and $\mathbf{b}_{i+1}$ projects to $\tilde{\mathbf{b}}_{i+1} + \mu_{i+1,i}\tilde{\mathbf{b}}_i$. So the Lovász criterion compares the norms of these two projected vectors and says that the second one is not much shorter than the first one, like the 2d case.

It is interesting to note that this condition is extremely local and it can be extended to looking at k vectors at a time to give a 2^k time, $2^{n/k}$ approximation to SVP (although we will not describe this extension here). We want to argue that finding an LLL-reduced basis is enough to get an approximate shortest vector.

Lemma 21. *Let $1/4 < \delta < 1$. If $\mathbf{B}$ is a δ-LLL reduced basis, then*

$$\|\mathbf{b}_1\| \leq \left(\frac{2}{\sqrt{4\delta - 1}}\right)^{n-1} \cdot \lambda_1$$

Proof. By the Lovász condition, we know that

$$\left\|\tilde{\mathbf{b}}_{i+1}\right\|^2 \geq (\delta - \mu_{i+1,i}^2)\left\|\tilde{\mathbf{b}}_i\right\|^2$$

Now, we use the fact that the basis $\mathbf{B}$ is size-reduced, that is $|\mu_{i+1,i}| \leq \frac{1}{2}$, to get

$$\left\|\tilde{\mathbf{b}}_{i+1}\right\| \geq \frac{\sqrt{4\delta - 1}}{2} \cdot \left\|\tilde{\mathbf{b}}_i\right\|$$

Thus, we have

$$\left\|\tilde{\mathbf{b}}_n\right\| \geq \frac{\sqrt{4\delta - 1}}{2} \cdot \left\|\tilde{\mathbf{b}}_{n-1}\right\| \geq \ldots \geq \left(\frac{\sqrt{4\delta - 1}}{2}\right)^{n-1} \cdot \left\|\tilde{\mathbf{b}}_1\right\| = \left(\frac{\sqrt{4\delta - 1}}{2}\right)^{n-1} \cdot \|\mathbf{b}_1\|$$

From this, together with the fact that $\lambda_1 \geq \min_i \left\|\tilde{\mathbf{b}}_i\right\|$, we infer that

$$\|\mathbf{b}_1\| \leq \left(\frac{2}{\sqrt{4\delta - 1}}\right)^{n-1} \cdot \min_i \left\|\tilde{\mathbf{b}}_i\right\| \leq \left(\frac{2}{\sqrt{4\delta - 1}}\right)^{n-1} \cdot \lambda_1$$

□

So, in any LLL-reduced basis, the first vector is a "good" approximation to the shortest vector. It is interesting to note that the LLL algorithm gives us quite a bit more. Indeed, we can use the fact that the i-th largest element of the set $\left\{\left\|\tilde{\mathbf{b}}_j\right\|\right\}_j$ gives us a lower bound on λ_i to get a comparable approximation on λ_i for all i.

7.3 Finding an LLL-reduced basis

Lemma 21 reduces the problem of getting a $2^{O(n)}$ approximation to SVP to finding an LLL-reduced basis. In this section we describe the LLL algorithm to find a reduced basis and analyze it.

Input Basis $\mathbf{b}_1, \mathbf{b}_2 \cdots \mathbf{b}_n$
while
1 Compute $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2 \cdots \tilde{\mathbf{b}}_n$
2 **for** $i = 2$ to n // Reduction Step
for $j = i - 1$ to 1
$\mathbf{b}_i \leftarrow \mathbf{b}_i - c_{i,j}\mathbf{b}_j$ where $c_{i,j} = \lfloor \mu_{i,j} \rceil$
3 **if** $\exists\, i$ such that $\left\|\tilde{\mathbf{b}}_{i+1}\right\| < \sqrt{\delta - \mu_{i+1,i}^2}\left\|\tilde{\mathbf{b}}_i\right\|$ // Swap step[a]
Swap $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$
else
Output $\mathbf{b}_1, \mathbf{b}_2 \cdots \mathbf{b}_n$

[a] We do not recompute the $\tilde{\mathbf{b}}_i$'s again because the Reduction step does not affect $\tilde{\mathbf{b}}_i$'s.

Figure 12: The LLL algorithm

The LLL-algorithm is an iterative algorithm where in each iteration, we first replace the vectors $\mathbf{b}_i$'s by "approximately" orthogonal vectors in the lattice obtained from the vectors $\tilde{\mathbf{b}}_i$'s. After this, we check if the Lovász criterion is violated for any pair of vectors $\mathbf{b}_i, \mathbf{b}_{i+1}$. In case of a violation, we swap the two vectors and continue.

Correctness. To prove correctness, we need to show that, if the algorithm terminates, then the output basis is LLL-reduced.

In order to show that $|\mu_{i,j}| \leq \frac{1}{2}$, consider the last iteration. It will not do any swaps and hence the vectors $\tilde{\mathbf{b}}_i$'s will remain unchanged in the iteration. On the other hand, because we subtract $c_{i,j}$'s, the values of $\mu_{i,j}$'s changes. Namely $\mu_{i,j}^{new} = \mu_{i,j}^{old} - \lfloor \mu_{i,j}^{old} \rceil$ and hence $\left|\mu_{i,j}^{new}\right| \leq 1/2$. Note that this relies on the fact that we are decrementing j in step 2 of the algorithm. The Lovász criterion is satisfied by termination – if it was not satisfied for some i, then we would swap $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$ and iterate again.

Number of Iterations The termination is a potential argument. We define a non-negative potential function $\phi(\mathbf{B})$ for any basis and then show that it was not too large to begin with and that each iteration reduces this function by a constant.

Let $\phi(\mathbf{B}) = \prod_i \phi_i(\mathbf{B})$ where

$$\phi_i(\mathbf{B}) = |\det(\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2 \cdots \mathbf{b}_i))|$$

Where the determinant for non full rank matrices is defined as $\det(\mathbf{B}) = \det(\sqrt{\mathbf{B}^\top \mathbf{B}})$. Another way to write it would be

$$\phi_i(\mathbf{B}) = \left\|\tilde{\mathbf{b}}_1\right\| \cdot \left\|\tilde{\mathbf{b}}_2\right\| \cdots \left\|\tilde{\mathbf{b}}_i\right\|$$

Lemma 22. *$\phi(\mathbf{B})$ is not too large to begin with*

$$\phi(\mathbf{B}_{init}) \leq \prod_{i=1}^{n} \|\mathbf{b}_1\| \cdot \|\mathbf{b}_2\| \cdots \|\mathbf{b}_i\| \leq \max_i(\|\mathbf{b}_i\|)^{O(n^2)}$$

So, $\log(\phi(\mathbf{B}_{init})) = \text{poly}(n, W)$ where W is the bit length of the vectors. We also know that $\phi(\mathbf{B}) \geq 1$ because of the fact that we are dealing with integer lattices and each potential ϕ_i can be interpreted as the i-dimensional volume enclosed by the vectors $\mathbf{b}_1, \mathbf{b}_2 \cdots \mathbf{b}_i$.

Lemma 23. *The reduction step does not change the potential function*

This is evident by looking at $\phi_i(\mathbf{B}) = \left\|\tilde{\mathbf{b}}_1\right\| \left\|\tilde{\mathbf{b}}_2\right\| \cdots \left\|\tilde{\mathbf{b}}_i\right\|$ and observing that the reduction step leaves $\tilde{\mathbf{b}}_i$'s invariant.

Lemma 24. *The swap step reduces ϕ by a constant factor.*

Proof. Let us say that $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$ were swapped. This only affects the value of $\phi_i(\mathbf{B})$ because changing order of vectors does not affect the determinant.

The old value of ϕ_i is, $\phi_i^{old} = \left\|\tilde{\mathbf{b}}_1\right\| \cdot \left\|\tilde{\mathbf{b}}_2\right\| \cdots \left\|\tilde{\mathbf{b}}_i\right\|$ while the new value is $\phi_i^{new} = \left\|\tilde{\mathbf{b}}_1^{new}\right\| \cdot \left\|\tilde{\mathbf{b}}_2^{new}\right\| \cdots \left\|\tilde{\mathbf{b}}_{i-1}^{new}\right\| \cdot \left\|\tilde{\mathbf{b}}_{i+1}^{new}\right\|$. We see that (a) $\mathbf{b}_j^{new} = \mathbf{b}_j$ for $j < i$ and (b) the component of $\mathbf{b}_{i+1}$ orthogonal to the span of $\mathbf{b}_1, \mathbf{b}_2 \cdots \mathbf{b}_{i-1}$ is $\tilde{\mathbf{b}}_{i+1} + \mu_{i+1,i}\tilde{\mathbf{b}}_i$. So,

$$\frac{\phi^{new}}{\phi^{old}} = \frac{\phi_i^{new}}{\phi_i^{old}} = \frac{\left\|\tilde{\mathbf{b}}_{i+1} + \mu_{i+1,i}\tilde{\mathbf{b}}_i\right\|}{\left\|\tilde{\mathbf{b}}_i\right\|} < \sqrt{\delta}$$

So as long as $\delta < 1$ is a fixed constant, we know that the potential function decreases by a constant factor. □

8 Applications of LLL

The LLL algorithm has numerous applications all through the fields of optimization, algorithm design, number theory and cryptography. The most famous, and perhaps the first, use of the LLL algorithm was to design a polynomial-time algorithm for factoring polynomials over the rationals [**?**]. LLL also gives us an algorithm for finding the exact shortest vector. While the algorithm takes exponential time, it was the first algorithm to solve exact SVP for a fixed dimension. There are many other applications including Coppersmith's method for finding small roots of polynomials over the integers with its ensuing applications to cryptanalysis, algorithms for integer programming, and approximate closest vector algorithms.

8.1 Computing the Shortest Vector in $2^{O(n^2)}$ Time

An immediate application of the LLL algorithm is a method of solving the shortest and closest vector problems *exactly* in n dimensions in time $2^{O(n^2)} \cdot \text{poly}(||\mathbf{B}||)$. We describe this method below, but first let us start with a neat property of LLL-reduced bases. This property says that the coefficients of any shortest vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ relative to an LLL-reduced basis $\mathbf{B}$ are not too large.

Lemma 25. *Let $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \ldots\ \mathbf{b}_n]$ be an LLL-reduced basis and let $\mathbf{v} = \sum_{i=1}^{n} c_i\mathbf{b}_i$ be any shortest vector in the lattice $\mathcal{L}(\mathbf{B})$. Then, for all i, $|c_i| = 2^{O(n)}$.*

Proof. Let $\widetilde{\mathbf{B}} = \tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n$ denote the Gram-Schmidt orthogonalization of $\mathbf{B}$. Write the shortest vector $\mathbf{v}$ in the terms of the GS basis as $\mathbf{v} = \sum_i c_i \mathbf{b}_i = \sum_i \tilde{c}_i \tilde{\mathbf{b}}_i$.

We look at the last i such that c_i is non-zero. Then $c_i = \tilde{c}_i$ because it is the only coefficient contributing in the $\tilde{\mathbf{b}}_i$ direction. Since $\|\mathbf{b}_1\| \geq \|\mathbf{v}\| \geq c_i \left\|\tilde{\mathbf{b}}_i\right\|$, we get that

$$c_i \leq \frac{\|\mathbf{b}_1\|}{\left\|\tilde{\mathbf{b}}_i\right\|} \leq \left(\frac{2}{\sqrt{4\delta - 1}}\right)^{i-1}$$

by Lemma 21. □

Given Lemma 25, a $2^{O(n^2)}$-time algorithm is immediate. Simply iterate over all $c_i \in [-2^{O(n)}, 2^{O(n)}]$, compute $\sum_{i=1}^{n} c_i \mathbf{b}_i$ and output the shortest one among them.

8.2 $2^{O(n)}$-approximate CVP in Polynomial Time

In this section, we present two algorithms to compute the approximately closest vector in a lattice. Both are due to Babai and both run in polynomial time. We will simply describe the first of these, called the "Rounding algorithm", without a detailed analysis. But we will analyze in detail the second, called the "Nearest Plane algorithm".

Babai's Rounding Algorithm. Given an LLL-reduced basis $\mathbf{B}$ and a target vector $\mathbf{t} \in \mathbb{Z}^n$, output $\mathbf{B} \cdot \lfloor \mathbf{B}^{-1}\mathbf{t} \rceil$ as the closest vector.

In words, first express the target vector $\mathbf{t}$ in terms of the basis $\mathbf{B}$. The resulting coefficients are not necessarily integers (if they were, $\mathbf{t}$ would already be in the lattice and we would be done), but one can of course round them to the nearest integers. This gives us the coefficients of the candidate closest lattice vector.

Lemma 26. *Let* $\mathbf{y} = \mathsf{Closest}_{\mathbf{B}}(\mathbf{t})$ *be the closest lattice vector to* $\mathbf{t}$. *Then, Babai's rounding algorithm outputs a vector* $\mathbf{z} \in \mathcal{L}(\mathbf{B})$ *such that* $\|\mathbf{t} - \mathbf{z}\| \leq C_n \cdot \|\mathbf{t} - \mathbf{y}\|$ *where* $C_n \leq 1 + 2n \cdot (9/2)^{n/2}$.

Exercises

1. Consider the basis
$$\mathbf{B} = \begin{pmatrix} 123 & 1 \\ 6764 & 55 \end{pmatrix}$$
   - Which of the following vectors belong to the lattice $\mathcal{L}(\mathbf{B})$?
$$\mathbf{v}_1 = \begin{pmatrix} 129 \\ 143 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 1/2 \\ 10 \end{pmatrix} \text{ and } \mathbf{v}_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$
   - What is the determinant of $\mathcal{L}(\mathbf{B})$?
   - Find the Gram-Schmidt orthogonalization of $\mathbf{B}$.
   - Find the shortest vector in $\mathcal{L}(\mathbf{B})$ (note that there may be many).
   - Find a shortest basis of $\mathcal{L}(\mathbf{B})$ (note that there may be many).

2. Let $\mathbf{e}_1, \ldots, \mathbf{e}_n$ be the n-dimensional unit vectors. Show that for every n-dimensional integer lattice $\mathcal{L}$ (namely, $\mathcal{L} \subseteq \mathbb{Z}^n$), the vectors $\det(L) \cdot \mathbf{e}_i \in \mathcal{L}$.

3. Given a basis $\mathbf{B}$, check if $\mathcal{L}(\mathbf{B})$ is a cyclic lattice, where a lattice $\mathcal{L}$ is called cyclic if for every lattice vector $\mathbf{x} \in \mathcal{L}$, any cyclic rotation of the coordinates of $\mathbf{x}$ is also in $\mathcal{L}$. For example, the lattice $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ where $\mathbf{b}_1 = (2,0,0)^T$, $\mathbf{b}_2 = (0,2,0)^T$ and $\mathbf{b}_3 = (1,1,1)^T$ is cyclic.

4. Describe a procedure that given any set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$, find a basis for the lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ (notice that these vectors are not necessarily linearly independent and that in particular, n might be greater than m). There is no need to analyze the running time. A corollary is that any set of vectors in $\mathbb{Z}^m$ spans a lattice.

5. Let $\mathcal{L}$ be a lattice. Recall that Minkowski's Convex Body Theorem states that any convex, centrally symmetric n-dimensional body S with $\mathsf{vol}(S) > 2n \cdot \det(\mathcal{L})$ contains a non-zero lattice point. Show that all the three conditions – convexity, central symmetry and the lower-bound on the volume – are necessary for this theorem to be true. Namely, for the lattice $\mathcal{L} = \mathbb{Z}^n$, show:
   - a convex set S_1 with $\mathsf{vol}(S_1) > 2n \cdot \det(\mathcal{L})$ that does not contain a lattice point. Note that S_1 has to be necessarily centrally asymmetric.
   - a centrally symmetric set S_2 with $\mathsf{vol}(S_2) > 2n \cdot \det(\mathcal{L})$ that does not contain a lattice point. Note that S_2 has to be necessarily non-convex.
   - a convex, centrally symmetric set S_3 with $\mathsf{vol}(S_3) = 2n \cdot \det(\mathcal{L})$ that does not contain a non-zero lattice point.

6. Despite lattices with much shorter vectors than predicted, Minkowski's theorem is tight for general lattices. In particular, there is a family of lattices $\{\mathcal{L}_n\}_{n \in \mathbb{N}}$ where $\mathcal{L}_n$ lives in n dimensions, and
$$\lambda_1(\mathcal{L}_n) \geq c \cdot \sqrt{n} \cdot \det(\mathcal{L}_n)^{1/n}$$
   where c is a universal constant independent of n.

   Show that such a family of lattices exists (your proof doesn't have to construct this family, you merely have to show existence).

7. (**) Same as the previous problem except show an explicit construction of such a family of lattices $\{\mathcal{L}_n\}_{n \in \mathbb{N}}$.