Lattice-Based Cryptography

KG

September 25, 2017

Contents

1 Introduction

In this note, we discuss the basic definitions and results about lattices, how lattices can be used in cryptography and how we could go about solving the lattice problems that arise from cryptographic use.

Many different mathematical concepts share the name *lattice*, but the one we are interested in is essentially integer linear combinations of linearly independent vectors in a real vector space. This notion of lattice arises naturally in many areas of mathematics, and also in many applications, leading to a rich and varied theory that was well-developed long before its use in cryptography.

Lattices have many uses in cryptography, and have for instance been used to attack many cryptographic constructions. There have been many attempts to construct cryptographic systems based on lattices, and some of these have failed. Recently, the theory of lattice-based cryptography has grown significantly, especially related to so-called *homormophic* cryptosystems.

However, in this note, we are not interested in using lattices to attack cryptosystem or these recent constructive developments, but rather the fact that there does not seem to be any fast quantum algorithms for solving difficult lattice-related problems. This makes lattice-based cryptography into a candidate for *quantum-safe* cryptography.

Section 2 gives a brief overview of the basic definitions regarding lattices. We need this basic theory to describe an extremely simple way to construct a lattice-based public key cryptosystem in Section 3. Finally, in Section 4 we discuss a fairly basic algorithm for searching for short vectors, whose performance can be significantly improved by first using the celebrated LLL algorithm.

This text is intended for a reader that is familiar with mathematical language, basic algebra (groups, rings, fields and linear algebra) and elementary computer science (algorithms).

This text is sometimes informal, in particular with respect to computational complexity. Every informal claim in this text can be made precise, but the technical details are out of scope for this note.

This text uses colour to indicate who is supposed to know what. When discussing cryptography, red denotes secret information known only by Alice or Bob. Blue denotes information that an eavesdropper will see. Information that is assumed to be known by both Alice and Bob (as well as Eve) is not coloured.

We also colour for theorems about computation, where blue denotes information that an algorithm gets as input and can use directly, while red denotes information that exists, but has to be computed somehow before it can be used directly. Information that is considered fixed (such as the specific group in use, group order, generator, etc.) is not coloured.

2 Lattices

There are several common, equivalent ways to define lattices. We have chosen a variant that is convenient for our purposes.

Definition 1. A *lattice* is a subgroup of $\mathbb{R}^n$ of the form

$$\Lambda = \mathbb{Z}\mathbf{b}_1 + \mathbb{Z}\mathbf{b}_2 + \cdots + \mathbb{Z}\mathbf{b}_r = \{\sum_{i=1}^{r} a_i\mathbf{b}_i \mid a_1, \ldots, a_r \in \mathbb{Z}\},$$

where $\mathbf{b}_1, \ldots, \mathbf{b}_r$ are linearly independent vectors in $\mathbb{R}^n$.

A *basis* for Λ is any set of linearly independent vectors $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{r'}$ such that $\Lambda = \mathbb{Z}\mathbf{c}_1 + \cdots + \mathbb{Z}\mathbf{c}_{r'}$.

From a basis $\mathbf{b}_1, \ldots, \mathbf{b}_r$ we get an $r \times n$ matrix B whose ith row is $\mathbf{b}_i$. Then

$$\Lambda = \{\mathbf{a}B \mid \mathbf{a} \in \mathbb{Z}^n\}.$$

The matrix B is called a basis matrix or just a basis. Since the rows of B are linearly independent, the rank of the matrix is equal to the number of rows.

Proposition 1. *Let Λ be a lattice, and let B and C be two bases for Λ. Then B and C have the same rank r, and there exists an $r \times r$ invertible integer matrices U such that $UB = C$ and U^{-1} is an integer matrix.*

Proof. Every row of B is in Λ which is a subset of the row space of C, so the row space of B is a subspace of the row space of C. The converse also holds, so the matrices have the same row space and therefore also the same rank.

For every row $\mathbf{b}_i$ in B, there exists an integer vector $\mathbf{a}_i \in \mathbb{Z}^r$ such that $\mathbf{b}_i = \mathbf{a}_i C$. Combining these expression, we get an $r \times r$ integer matrix U such that $B = UC$. In the same way, we get an $r \times r$ integer matrix V such that $C = VB$, and $B = UVB$.

Since B and C have maximal rank (and therefore have right-inverses) U and V must be inverses. □

Definition 2. The *rank* of a lattice is the number of vectors in a basis. If Λ in $\mathbb{R}^n$ has rank n, we say that Λ is a *full rank* lattice.

2.1 The fundamental domain

Let $\mathbf{b}_1, \ldots, \mathbf{b}_r$ be a basis for a lattice Λ. The *fundamental domain* of the lattice is the parallelepiped

$$\left\{ \sum_{i=1}^{r} a_i \mathbf{b}_i \mid 0 \le a_i < 1 \right\}.$$

When the lattice has full rank, it can be shown that the volume of the fundamental domain is $|\det(B)|$. When $r < n$, we define the volume of the fundamental domain to be $\sqrt{|\det(BB^T)|}$.

Since any two bases are related by an invertible integer matrix, which has determinant ± 1, it is clear that the volume of the fundamental domain is independent of the choice of basis. We call this volume the *determinant* or the *volume* of the lattice, denoted by $\det(\Lambda)$, or sometimes by $\det(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r)$.

The Gram-Schmidt algorithm takes a vector space basis $\mathbf{b}_1, \ldots, \mathbf{b}_r$ as input and constructs an orthogonal basis $\mathbf{b}_1^*, \ldots, \mathbf{b}_r^*$ recursively, beginning with $\mathbf{b}_1^* = \mathbf{b}_1$ and then computing

$$\mu_{ij} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}, \quad 1 < i \le r, 1 \le j < i, \qquad \text{and} \qquad \mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \mathbf{b}_j^*. \tag{1}$$

Since a determinant is unchanged by elementary row operations, if $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is a basis for a full-rank lattice Λ, then

$$\det(\Lambda) = \prod_i \|\mathbf{b}_i^*\| \le \prod \|\mathbf{b}_i\|.$$

The value

$$\frac{\prod \|\mathbf{b}_i\|}{\det(\Lambda)}$$

is called the *orthogonality defect* of the basis.

E *Exercise* 1. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r \in \mathbb{R}^n$ be a lattice basis. Show that we can compute the Gram-Schmidt basis $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_r^*$ and coefficients μ_{ij}, $1 \le j < i \le r$ using $2nr^2 - 1$ arithmetic operations.

E *Exercise* 2. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r$ be a basis for a lattice Λ, let B be the corresponding matrix, and let $\mathbf{b}_1^*, \ldots, \mathbf{b}_r^*$ be the corresponding Gram-Schmidt orthogonal basis.

a. Show that $\mathbf{b}_1^*, \ldots, \mathbf{b}_r^*$ can be extended to an orthogonal basis $\mathbf{b}_1^*, \ldots, \mathbf{b}_r^*, \mathbf{c}_1^*, \ldots, \mathbf{c}_{n-r}^*$ for $\mathbb{R}^n$.

b. Let $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r$ be an orthonormal basis for $\mathbb{R}^r$. Let P be the linear map from $\mathbb{R}^n$ to $\mathbb{R}^r$ that takes $\mathbf{b}_i^*$ to $\|\mathbf{b}_i^*\|\mathbf{e}_i$, $1 \leq i \leq r$, and $\mathbf{c}_i^*$ to $\mathbf{0}$, $1 \leq i \leq n-r$. Show that for any vector $\mathbf{v} \in \operatorname{span}\{\mathbf{b}_1^*, \ldots, \mathbf{b}_r^*\}$, $\|\mathbf{v}\| = \|\mathbf{v}P\|$.

c. Show that the image ΛP of Λ under P is a full-rank lattice, and $\mathbf{b}_1 P, \mathbf{b}_2 P, \ldots, \mathbf{b}_r P$ is a basis for ΛP.

d. Show that $\det(\Lambda) = \det(\Lambda P)$.

e. Show that $\det(\Lambda) = \prod_{i=1}^{r} \|\mathbf{b}_i^*\|$.

Another interesting property of the fundamental domain of a full-rank lattice is that any point in space can be expressed uniquely as the sum of a lattice point and a point in the fundamental domain.

E *Exercise* 3. Let Λ be a lattice with basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ and let $\mathbf{z} \in \mathbb{R}^n$. Show that unique integers $a_1, a_2, \ldots, a_n$ and real numbers $\alpha_1, \alpha_2, \ldots, \alpha_n \in [0, 1)$ exist such that

$$\mathbf{z} = \sum_i a_i \mathbf{b}_i + \sum_i \alpha_i \mathbf{b}_i.$$

2.2 Short vectors

From now on, we shall assume that we only work with full rank lattices.

Definition 3. Let Λ be a lattice. The *ithe successive minimum* $\lambda_i(\Lambda)$ is the minimal real number such that there are i linearly independent vectors of length at most $\lambda_i(\Lambda)$ in Λ.

It is immediately clear that there are n successive minima, that $0 < \lambda_1(\Lambda) \leq \lambda_2(\Lambda) \leq \cdots \leq \lambda_n(\Lambda)$, and that $\lambda_1(\Lambda)$ is the shortest length of a non-zero vector in Λ. (Note that $\lambda_2(\Lambda)$ does not have to be the second shortest length of a non-zero vector.)

Fact 2. *There exists a constant γ_n, depending only on n, such that for any lattice Λ*

$$\lambda_1(\Lambda)^2 < \gamma_n \det(\Lambda)^{2/n}.$$

A natural question related to lattices is to ask what $\lambda_1(\Lambda)$ is, or if a lattice vector of length $\lambda_1(\Lambda)$ can be found.

Definition 4. The *shortest vector problem* for a lattice Λ is to find a vector $\mathbf{x} \in \Lambda$ such that $\|\mathbf{x}\| = \lambda_1(\Lambda)$.

Sometimes, there will be more than one non-zero vector of minimal length, so the shortest vector problem may not have a unique answer. For some applications, merely finding a short vector is sufficient.

Definition 5. The *γ-approximate shortest vector problem* for a lattice Λ is to find a vector $\mathbf{x} \in \Lambda$ such that $\|\mathbf{x}\| \leq \gamma\lambda_1(\Lambda)$.

A slightly different problem is to find a lattice vector close to some given point. It also has an approximate version.

Definition 6. The *closest vector problem* for a lattice $\Lambda \subseteq \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^n$ is to find $\mathbf{x} \in \Lambda$ such that for any $\mathbf{y} \in \Lambda$, $\|\mathbf{x} - \mathbf{z}\| \leq \|\mathbf{y} - \mathbf{z}\|$.

The *γ-approximate closest vector problem* for a lattice $\Lambda \subseteq \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^n$ is to find $\mathbf{x} \in \Lambda$ such that for any $\mathbf{y} \in \Lambda$, $\|\mathbf{x} - \mathbf{z}\| \leq \gamma\|\mathbf{y} - \mathbf{z}\|$.

An exhaustive search for short or close vectors will quickly become infeasible as the lattice dimension grows. There is evidence that these lattice problems are hard in a very fundamental way. However, the hardness depends on the lattice and on how the lattice is described.

E *Exercise* 4. Suppose Λ has a given orthogonal basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$. Show how to find a shortest vector in Λ and all the successive minima, essentially without any arithmetic.

If the basis is nearly orthogonal, the same approach as in the exercise will find short vectors and make it easier to find a shortest vector.

E *Exercise* 5. Let B be an invertible $n \times n$ matrix and let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ correspond to the n rows of B. Let $\mathbf{z} \in \mathbb{R}^n$ and let

$$(\alpha_1, \alpha_2, \ldots, \alpha_n) = \mathbf{z}B^{-1}.$$

Show that

$$\mathbf{z} = \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 + \cdots + \alpha_n\mathbf{b}_n.$$

For the following exercise, it is convenient to introduce the following notation: $\lfloor\alpha\rceil$ is the nearest even integer to α, and for a vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$,

$$\lfloor\boldsymbol{\alpha}\rceil = (\lfloor\alpha_1\rceil, \lfloor\alpha_2\rceil, \ldots, \lfloor\alpha_n\rceil).$$

E *Exercise* 6. Suppose Λ has a given orthogonal basis B, and let $\mathbf{z} \in \mathbb{R}^n$. Suppose $\mathbf{a} = \lfloor\mathbf{z}B^{-1}\rceil$. Explain why $\mathbf{a}B$ is the closest vector in Λ to $\mathbf{z}$.

If the basis is nearly orthogonal, the same approach as in the exercise will tend to find the closest vector if $\mathbf{z}$ was reasonably close to a lattice point.

3 GGH

One idea for a symmetric encryption scheme based on lattices is to have a lattice Λ with a nearly orthogonal basis B as a secret key. To encrypt we somehow encode the message as a lattice vector $\mathbf{x} \in \Lambda$ and then add *random noise* $\mathbf{r}$ to that vector to get a

ciphertext $\mathbf{z} = \mathbf{x} + \mathbf{r}$. To decrypt, we can use our nearly orthogonal basis to find the closest vector $\mathbf{x}$ to $\mathbf{z}$ (using the technique from Exercise 6), and then decode to recover the message.

It is clear that we need to limit the magnitude of the random noise, since if it is too big, we will no longer be able to recover $\mathbf{x}$ as the closest vector. This could happen because $\mathbf{x}$ is no longer the closest vector to $\mathbf{z}$, or because our basis is not orthogonal and so does not perfectly solve the closest vector problem).

E *Exercise* 7. Let B be a basis for a lattice Λ. Show that with $\mathbf{x} \in \Lambda$ and $\mathbf{z} = \mathbf{x} + \mathbf{r}$, then $\lfloor \mathbf{z}B^{-1} \rceil B = \mathbf{x}$ if and only if $\lfloor \mathbf{r}B^{-1} \rceil = \mathbf{0}$.

E *Exercise* 8. Recall that for a real vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, we have the norms $\|\boldsymbol{\alpha}\|_1 = \sum_i |\alpha_i|$ and $\|\boldsymbol{\alpha}\|_\infty = \max_i |\alpha_i|$.

Let B be a basis for a lattice Λ, let ρ be a bound on the $\|\cdot\|_1$ norm of the columns of B^{-1}. Show that for any vector $\mathbf{r}$, we have that

$$\|\mathbf{r}B^{-1}\|_\infty \leq \rho \|\mathbf{r}\|_\infty.$$

Explain how this can be used to find a bound on the random noise when encrypting, to ensure decryption still works.

It is tempting to turn this idea into a public key encryption scheme by publishing a basis for the lattice. Obviously, we cannot publish our nearly orthogonal basis B, since this is essentially the decryption key.

Recall that any lattice Λ with basis matrix B, if U is an integer matrix with determinant ± 1, then $C = UB$ is another basis matrix for Λ.

One idea is then to create and publish a different basis for the lattice, one that is not nearly orthogonal, and therefore cannot be used to find the closest vector using the approach from Exercise 6.

As usual, while we could attempt to embed the message in the vector $\mathbf{x}$, it makes more sense to use $\mathbf{x}$ and $\mathbf{r}$ as keys for a symmetric cryptosystem.

The public key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ is based on lattices in $\mathbb{R}^n$ and a symmetric cryptosystem $(\mathbb{R}^n \times \mathbb{R}^n, \mathcal{P}, \mathcal{C}, \mathcal{E}_s, \mathcal{D}_s)$.

- The *key generation* algorithm $\mathcal{K}$ chooses a lattice Λ by choosing a basis matrix B that is nearly orthogonal. It chooses an integer matrix U with determinant ± 1 and computes $C = UB$. It then outputs $ek = C$ and $dk = B$.

- The *encryption* algorithm $\mathcal{E}$ takes as input an encryption key $ek = C$ and a message $m \in \mathcal{P}$. It chooses a random vector $\mathbf{a} \in \mathbb{Z}^n$ and random noise $\mathbf{r}$. Then it computes $\mathbf{x} = \mathbf{a}C$, $\mathbf{z} = \mathbf{x} + \mathbf{r}$, and encrypts the message as $w = \mathcal{E}_s((\mathbf{x}, \mathbf{r}), m)$. It outputs the ciphertext $c = (\mathbf{z}, w)$.

- The *decryption* algorithm $\mathcal{D}$ takes as input a decryption key $dk = B$ and a ciphertext $c = (\mathbf{z}, w)$. It computes $\mathbf{x} = \lfloor \mathbf{z}B^{-1} \rceil B$ and $\mathbf{r} = \mathbf{z} - \mathbf{x}$, and decrypts the message as $m = \mathcal{D}_s((\mathbf{x}, \mathbf{r}), w)$. If $\mathcal{D}_s$ outputs the special symbol $\perp$ indicating decryption failure, then $\mathcal{D}$ outputs $\perp$, otherwise it outputs m.

E *Exercise* 9. The above is an informal description of a public key encryption scheme. Write down carefully (except for how to choose B, U, $\mathbf{a}$ and $\mathbf{r}$) what the three algorithms $\mathcal{K}$, $\mathcal{E}$ and $\mathcal{D}$ are. Show that the triple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a public key encryption scheme.

4 Finding closest vectors

4.1 Enumerating short vectors

We would like to find a short vector in a lattice. One idea would simply be to enumerate all linear combinations of the basis vectors with some bound on the coefficients. Unfortunately, short vectors could in principle come from linear combinations with large coefficients. Instead, we shall use the Gram-Schmidt basis to bound the size of the coefficients.

We shall find all points $\mathbf{x}$ in a lattice with $\|\mathbf{x}\|^2 \leq A^2$ for some bound A^2.

Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be a basis for Λ, and let $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ be the corresponding Gram-Schmidt basis. Note that for any vector $\mathbf{x} \in \Lambda$, we can write it as a linear combination of the Gram-Schmidt basis vectors $\mathbf{x} = \sum_i \alpha_i \mathbf{b}_i^*$ and then

$$\|\mathbf{x}\|^2 = \sum_{i=1}^{n} \alpha_i^2 \|\mathbf{b}_i^*\|^2.$$

Recall that $\mathbf{b}_n^*$ is the part of $\mathbf{b}_n$ that is orthogonal to all the earlier basis vectors. This means when $\mathbf{x} = \sum_i a_i \mathbf{b}_i = \sum_i \alpha_i \mathbf{b}_i^*$, then $a_n = \alpha_n$. Therefore, if $\alpha \|\mathbf{b}_n^*\| > A$ then $\|\mathbf{x}\| > A$.

We therefore begin by enumerating vectors with n-th coordinate a_n between $-\lfloor A/\|\mathbf{b}_n^*\| \rfloor$ and $+\lfloor A/\|\mathbf{b}_n^*\| \rfloor$.

Given a_n, we can now consider the possibilities for a_{n-1}. Of course, this time, the contribution in the direction of $\mathbf{b}_{n-1}^*$ is that given by $a_{n-1}\mathbf{b}_{n-1}$ and $a_n \mathbf{b}_n$, where the latter's contribution is $a_n \mu_{n,n-1} \|\mathbf{b}_{n-1}^*\|$. So given a_n, we want to enumerate all the $(n-1)$-th coordinates a_{n-1} such that

$$(a_{n-1} + a_n \mu_{n,n-1})^2 \|\mathbf{b}_{n-1}^*\|^2 + a_n^2 \|\mathbf{b}_{n-1}^*\|^2 \leq A^2.$$

In general, given $a_{i+1}, \ldots, a_n$, we consider the possibilities for a_i. Again, we want to enumerate all a_i such that

$$(a_i + \sum_{j=i+1}^{n} a_j \mu_{ji})^2 \|\mathbf{b}_i^*\|^2 + \sum_{j=i+1}^{n} (a_j + \sum_{k=j+1}^{n} a_k \mu_{k,j})^2 \|\mathbf{b}_j^*\|^2 \leq A^2.$$

For some choices of $a_{i+1}, \ldots, a_n$ there may be no possible choices for a_i, in which case we stop and continue with other choices for $a_{i+1}, \ldots, a_n$.

Whenever we find a non-empty region for a_1 and enumerate those values, we enumerate lattice vectors of length less than A. It is clear that for any lattice point $\mathbf{x}$ of length less than A, this point must be among the lattice point eventually enumerated.

This enumeration algorithm must enumerate a large number of possible coordinates. For the ith coordinate, we can upper-bound the number of possibilities for a_i by $A/\|\mathbf{b}_i^*\|$, so the total number of coordinates enumerated is bounded by

$$\prod_{i=1}^{n} \frac{A}{\|\mathbf{b}_i^*\|} = \frac{A^n}{\det(\Lambda)}.$$

We can also solve the closest vector problem if we can first find an estimate for the closest vector and a reasonable upper bound on how far the estimate is from the closest point. Given this, we can enumerate all the short vectors and thereby all the lattice points close to the estimate, one of which must be closest.

There are faster algorithms for finding a shortest or closest vector.

4.2 LLL Algorithm

As we saw, if we have an orthogonal basis, we can solve the closest vector problem, and if we have a nearly orthogonal basis, we can solve closest vector problem if the closest vector is close enough to a lattice point.

The natural question is how to find a reasonably good basis that will allow us to solve the closest vector problem. The first goal should be to be precise about what we mean by "reasonably good".

Definition 7. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be a lattice basis, with corresponding orthogonal basis $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ and Gram-Schmidt coefficients μ_{ij} for $1 \le j < i \le n$, as defined in (1). Let $\frac{1}{4} < \delta < 1$ be a real number. We say that the basis is *δ-LLL-reduced* if

$$|\mu_{ij}| \le \frac{1}{2} \qquad \text{for all } 1 \le j < i \le n, \text{ and} \tag{2}$$

$$\delta\|\mathbf{b}_{i-1}^*\|^2 \le \|\mathbf{b}_i^*\|^2 + \mu_{i,i-1}^2\|\mathbf{b}_{i-1}^*\|^2 \qquad \text{for all } 2 \le i \le n. \tag{3}$$

When a basis satisfies (2) we cannot easily make the basis vectors more orthogonal. When the basis satisfies (3), the basis vectors of the Gram-Schmidt orthogonalization will be ordered roughly according to length.

E *Exercise* 10. A common choice for δ is $3/4$. Show that in this case (3) becomes

$$\|\mathbf{b}_{i-1}^*\|^2 \le 2\|\mathbf{b}_i^*\|^2 \text{ for all } 2 \le i \le n.$$

Hint: You may use the fact that (2) also must hold.

That an LLL-reduced basis is somehow a good basis can be seen from the following fact, which we state without proof.

Fact 3. *Suppose $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ with corresponding basis matrix B is a $3/4$-LLL-reduced basis for a lattice Λ. Then $\|\mathbf{b}_1\| \le 2^{(n-1)/2}\lambda_1(\Lambda)$. Also, if $\mathbf{z} \in \mathbb{R}^n$, then*

$$\|\mathbf{z} - \lfloor \mathbf{z}B^{-1} \rceil B\| \le (1 + 2n(9/2)^{n/2})\|\mathbf{z} - \mathbf{x}\| \text{ for any } \mathbf{x} \in \Lambda.$$

We know that $\lambda_1(\Lambda) \leq \sqrt{\gamma}\det(\Lambda)^{1/n}$, which means that if we have an LLL-reduced basis and use $\|\mathbf{b}_1\|$ as our search bound, the enumeration approach from the previous section will have to enumerate at most

$$\frac{(2^{(n-1)/2}\gamma^{1/2}\det(\Lambda)^{1/n})^n}{\det(\Lambda)} = 2^{n(n-1)/2}\gamma^{n/2}.$$

While it does not affect the upper bound we deduced, having the Gram-Schmidt vectors not too small will decrease the total number of points the algorithm will iterate over.

We also note that the LLL-reduced basis will give us an estimate for the closest vector problem. (There are better ways to use the LLL-reduced basis.)

Having an LLL-reduced basis is therefore very useful. The question is how to find an LLL-reduced basis.

We will now discuss two ways to modify a lattice basis. The first is to use the initial basis vectors to modify the later basis vectors. This clearly results in a new basis for the same lattice, but the new basis will have the same Gram-Schmidt orthogonalization.

E *Exercise* 11. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be a lattice basis, and let $\mathbf{c}_1, \ldots, \mathbf{c}_n$ be another basis for the lattice defined by

$$\mathbf{c}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \alpha_{ij}\mathbf{b}_j,\ \alpha_{ij} \in \mathbb{Z}. \tag{4}$$

Let $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ and $\mathbf{c}_1^*, \ldots, \mathbf{c}_n^*$ be the corresponding Gram-Schmidt orthogonalization. Show that $\mathbf{b}_i^* = \mathbf{c}_i^*$ for $i = 1, 2, \ldots, n$.

E *Exercise* 12. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be a lattice basis, let $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*$ be the Gram-Schmidt orthogonalization with coefficients μ_{ij}, $1 < i \leq n$.

Suppose $|\mu_{ij}| \leq 1/2$ for $1 \leq j < i < k$ for some $k \leq n$. Define $\mathbf{b}_k^{(i)}$ for $1 \leq i \leq k$ by $\mathbf{b}_k^{(k)} = \mathbf{b}_k$ and

$$\mathbf{b}_k^{(i)} = \mathbf{b}_k^{(i+1)} - \left\lfloor \frac{\langle \mathbf{b}_k^{(i+1)}, \mathbf{b}_i^* \rangle}{\langle \mathbf{b}_i^*, \mathbf{b}_i^* \rangle} \right\rceil \mathbf{b}_i.$$

Consider now the new basis $\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}, \mathbf{b}_k^{(1)}, \mathbf{b}_{k+1}, \ldots, \mathbf{b}_n$.

a. Show that the new basis has the same Gram-Schmidt orthogonalization as the old basis.

b. Let μ'_{ij} be the Gram-Schmidt coefficients of the new basis. Show that $|\mu'_{ij}| \leq 1/2$ for $1 \leq j \leq i \leq k$.

a. Show that the above procedure essentially gives us an algorithm that for any lattice basis computes a new, equivalent basis with the same Gram-Schmidt orthogonalization that also satisfies (2).

b. Show that the resulting algorithm will compute the new basis using at most $2n^3$ arithmetic operations.

This result tells us that we can always modify a lattice basis such that (2) holds without changing the Gram-Schmidt orthogonalization. We can also do this relatively quickly.

Next, we want to modify our basis by changing the order of the basis vectors. Unlike the previous modification, this will change the resulting Gram-Schmidt orthogonal basis.

Before we describe and analyse this change, we need to define a new kind of volume for lattices that weights the basis vectors differently. For a basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ with corresponding Gram-Schmidt orthogonalization $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*$, define

$$d(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \prod_{i=1}^{n} \prod_{j=1}^{i-1} \|\mathbf{b}_j^*\| = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|^{n-i+1}.$$

Suppose $\mathbf{b}_1, \ldots, \mathbf{b}_n$ satisfies (2), but not (3), and i is the first index where the latter condition fails. Now suppose we change the order of the ith and $i+1$th basis vectors, so instead of considering the basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_i, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n$, we instead consider the basis $\mathbf{c}_1, \ldots, \mathbf{c}_n$ given by

$$\mathbf{c}_j = \begin{cases} \mathbf{b}_{i+1} & j = i \\ \mathbf{b}_i & j = i+1 \\ \mathbf{b}_j & \text{otherwise.} \end{cases}$$

E *Exercise* 13. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ and $\mathbf{c}_1, \ldots, \mathbf{c}_n$ be as above, and let $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ and $\mathbf{c}_1^*, \ldots, \mathbf{c}_n^*$ be the corresponding Gram-Schmidt orthogonalizations. Let μ_{ij} be the Gram-Schmidt coefficients for $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$.

a. Show that $\mathbf{b}_j^* = \mathbf{c}_j^*$ when $j \neq i, i+1$.

b. Show that $\mathbf{c}_i^* = \mathbf{c}_{i+1}^* + \mu_{i+1,i}\mathbf{b}_i^*$.

c. Show that

$$\prod_{j=1}^{k} \|\mathbf{b}_j^*\| = \prod_{j=1}^{k} \|\mathbf{c}_j^*\|$$

when $i \neq k$.

d. Show that

$$\frac{d(\mathbf{c}_1, \ldots, \mathbf{c}_n)}{d(\mathbf{b}_1, \ldots, \mathbf{b}_n)} \leq \sqrt{\delta}.$$

E *Exercise* 14. Show that for any lattice $\Lambda \in \mathbb{Z}^n$, then regardless of basis, there is a lower bound to $d(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ that is strictly larger than zero.

We are now ready to prove that a basis satisfying (2) and (3) can be computed relatively quickly.

Theorem 4. *Let B be a basis for a lattice $\Lambda \in \mathbb{Z}^n$. Then a δ-LLL-reduced basis C can be computed using less than $10n^3 \log d(B) / \log \delta^{-1}$ arithmetic operations.*

Proof. We begin by constructing a sequence of lattice bases $B_1, B_2, \ldots$ as follows.

The initial basis is $B_1 = B$.

We construct the $(k+1)$th basis from the kth basis B_k using the following two steps.

1. Use the algorithm from Exercise 12 to create a basis B'_k.
2. If the basis B'_k does not satisfy (3), and i is the first index where this condition fails we change the order of the ith and the $(i+1)$th basis vectors.

It is clear that if B_k satisfies (2) and (3), then $B_{k+1} = B_k$.

Furthermore, unless we changed the order of two basis vectors when creating B_{k+1}, then B_{k+1} satisfies (2) and (3).

Every time we do change the order of two basis vectors, Exercise 13 says that

$$\frac{d(B_{k+1})}{d(B_k)} \leq \sqrt{\delta},$$

from which we get that

$$d(B_{k+1}) \leq \delta^{k/2} d(B_1).$$

Since $d(B_{k+1}) \geq 1$, it follows that when $k > 2 \log d(B_1) / \log \delta^{-1}$ there can be no more changes of order, and we have computed a δ-LLL-reduced basis.

By Exercise 12 the first step requires at most $2n^3$ arithmetic operations. To do the second step, we need the Gram-Schmidt coefficients which we can compute using at most $2n^3$ arithmetic operations by Exercise 1. Finally, we can find the first index for which (3) does not hold using less than $3n^2$ arithmetic operations. This means that we can compute B_{k+1} from B_k using less than $5n^3$ arithmetic operations. The claim follows. □

E *Exercise* 15. The proof of Theorem 4 essentially describes an algorithm for computing an LLL-reduced basis for a lattice. Write out this algorithm carefully and restate Theorem 4 as a statement about the algorithm's time complexity (in terms of arithmetic operations, ignoring any other form of computation involved).

We must make two remarks about this theorem and its proof. The first remark is about measuring complexity in terms of arithmetic operations. Often, this is safe, but in this case we will need to do arithmetic with rational numbers, and in some computations the size (number of digits) of the rational numbers involved will increase exponentially, even when the number of operations is limited. Fortunately, it can be proved that the size of the rational numbers involved do not grow too badly, although it significantly affects the time required to run the algorithm.

The second remark is about the implied algorithm, which is extremly inefficient. In practice, there is no need to constantly recompute all the Gram-Schmidt coefficients. In other words, there is significant scope for optimization in the algorithm. Also, the proof significantly overestimates the cost of the implied algorithm.