F.J. Hobrecker

Solving the Bounded Distance Decoding Problem

A Generalization of Barnes-Wall Lattice Decoding and Its Application to Tensor Hexagonal Lattices

Bachelor Thesis

April, 2025

Thesis Supervisors: **prof. dr. L. Ducas**
prof. dr. ir. N. Mentens

Leiden University
Mathematical Institute and
Leiden Institute of Advanced Computer Science (LIACS)

Abstract

In this thesis, we explore the bounded distance decoding (BDD) problem and efficient solutions for the problem within specific lattice structures. The lattices of interest are the Barnes-Wall lattices and the tensor hexagonal lattices. For the Barnes-Wall lattices, we implement an efficient decoding algorithm based on the work of Micciancio and Nicolosi [MN08]. We then generalize this approach to develop a framework for solving the BDD problem in lattices of dimension $n = 2^\ell$ with $\ell \in \mathbb{Z}_{\geq 1}$, relying on certain hypotheses to ensure correct decoding. Finally, we apply this framework to the tensor hexagonal lattices. Our results show that efficient solutions for the BDD problem extend beyond the Barnes-Wall lattices.

Contents

Chapter 1

Introduction

The rapid advancements in quantum computing pose a serious threat to the security of many existing cryptographic systems. Quantum algorithms, such as Shor's algorithm, can efficiently solve the computationally hard problems on which they are based [FA24]. This threat has lead to the exploration of quantum-resistant cryptographic systems. A promising candidate is lattice-based cryptography, which relies on the hardness of certain lattice problems. One of these hard lattice problems is the bounded distance decoding (BDD) problem. It involves finding the closest lattice point to a given target point, provided that the target is within a certain bounded distance from the lattice.

This thesis investigates efficient algorithms for solving the BDD problem in certain lattice structures. We first introduce some fundamental concepts in lattice theory and formally define the BDD problem in Chapter 2. In Chapter 3 we examine two lattice families of interest: the Barnes-Wall lattices and the tensor hexagonal lattices. These lattices were chosen for their recursive constructions, which can be exploited to develop efficient decoding algorithms. Chapter 4 focuses on efficient decoding algorithms. We begin with the decoding algorithm for Barnes-Wall lattices, based on the work of Micciancio and Nicolosi [MN08]. We implement this algorithm and test its performance outside its unique decoding radius. We then generalize this approach to develop a framework for solving the BDD problem in lattices of dimension $n = 2^\ell$ with $\ell \in \mathbb{Z}_{\geq 1}$, relying on certain hypotheses to ensure correct decoding. Lastly, we show how one can apply this generalized algorithm to tensor hexagonal lattices.

Our results show that we can generalize the algorithm for solving the BDD problem in Barnes-Wall lattices to work for a broader class of lattices, including the tensor hexagonal lattices.

Chapter 2

Preliminary lattice theory

In this chapter, we introduce key concepts and definitions related to lattices and lattice problems.

Matrices are written in uppercase bold $\mathbf{B}$, vectors are written in lowercase bold $\boldsymbol{x}$ and are interpreted as row vectors, and scalars are written in normal lowercase λ. We use $\mathbb{R}$ for the real numbers, $\mathbb{Z}$ for the integers, $\mathbb{C}$ for the complex numbers and $\mathbb{G}$ for the Gaussian integers.

2.1 Lattices

A lattice is a collection of points in n-dimensional space that follows a repeating and regular pattern. Formally, we have the following definition:

Definition 1 (Real lattice, [DD18a])**.** *A real k-dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$. The lattice is said to be full rank if $k = \dim(\mathcal{L}) = n$.*

To clarify the components of this definition, we formally define additive and discrete subgroups.

Definition 2 (Additive subgroup, [Cor20])**.** *A subset $\mathcal{L} \subseteq \mathbb{R}^n$ is an additive subgroup if it satisfies:*

- $\mathbf{0} \in \mathcal{L}$,
- $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{L}$, $\boldsymbol{x} + \boldsymbol{y} \in \mathcal{L}$ *(closure under addition),*
- $\forall \boldsymbol{x} \in \mathcal{L}$, $-\boldsymbol{x} \in \mathcal{L}$ *(closure under negation).*

Definition 3 (Discrete subgroup, [DD18a, HWL08])**.** *A subset $\mathcal{L} \subseteq \mathbb{R}^n$ is discrete if the induced topology on $\mathcal{L}$ is discrete, i.e. every subset of $\mathcal{L}$ is open. In other words, for every $\boldsymbol{x} \in \mathcal{L}$, there exists a radius $\epsilon > 0$, such that the open ball*

$$\mathcal{B}(\boldsymbol{x}, \epsilon) = \{\boldsymbol{y} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{y}\| < \epsilon\}$$

contains no other points of $\mathcal{L}$, meaning $\mathcal{B}(\boldsymbol{x}, \epsilon) \cap \mathcal{L} = \{\boldsymbol{x}\}$.

Discreteness ensures that each lattice point has a neighborhood containing no other lattice points, thereby guaranteeing a strictly positive minimum distance between any to distinct lattice points [vW23]. Formally, we have the following:

Proposition 4 ([DD18a]). *Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a non-trivial additive subgroup of $\mathbb{R}^n$. Then $\mathcal{L}$ is discrete if and only if there exists a strictly positive and well-defined minimum distance between any two distinct points in $\mathcal{L}$.*

The additive structure of the lattice implies that the difference between any two distinct lattice points is itself a lattice point. This leads to the following definition:

Definition 5 (Minimum distance, [MG02]). *The minimum distance of a lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is given by:*

$$\lambda_1(\mathcal{L}) = \min_{\boldsymbol{x} \neq \boldsymbol{y} \in \mathcal{L}} \|\boldsymbol{x} - \boldsymbol{y}\| = \min_{\boldsymbol{x} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\boldsymbol{x}\|,$$

where $\lambda_1(\mathcal{L}) > 0$ and $\|\cdot\|$ is the Euclidean norm.

Every lattice can be described by a set of linearly independent generators (basis matrix). Throughout this thesis, we work with row notation, where the lattice is generated by the rows of the basis matrix. Formally, we have the following:

Definition 6 (Basis, [MG02]). *Let $\mathbf{B} \in \mathbb{R}^{k \times n}$ be a matrix with as rows k linearly independent vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_k \in \mathbb{R}^n$. The lattice generated by $\mathbf{B}$ is the set*

$$\mathcal{L}(\mathbf{B}) = \left\{ \boldsymbol{z} \cdot \mathbf{B} : \boldsymbol{z} \in \mathbb{Z}^k \right\} = \left\{ \sum_{i=1}^{k} z_i \boldsymbol{b}_i : z_i \in \mathbb{Z} \right\}$$

of all integer linear combinations of the rows of $\mathbf{B}$. The rank of the lattice is k and the dimension of the ambient space is n. When $k = n$, then $\mathcal{L}(\mathbf{B})$ is a full rank lattice.

From now on we assume that all lattices are full rank lattices with dimension n.

Definition 7 (Determinant, [DD18b]). *Let $\mathcal{L}$ be an n-dimensional lattice generated by some basis $\mathbf{B} \in \mathbb{R}^{n \times n}$. The determinant of the lattice is given by:*

$$\det(\mathcal{L}(\mathbf{B})) = \sqrt{\det\left(\mathbf{B}^\top \mathbf{B}\right)} = |\det(\mathbf{B})|.$$

Definition 8 (Automorphism group, [vP16]). *The automorphism group of a lattice $\mathcal{L}$ denoted $Aut(\mathcal{L})$, is the set of distance-preserving linear transformations of the space that fix the origin and take the lattice to itself.*

Definition 9 (Complex lattice, [For88, Cor20]). *A complex n-dimensional lattice $\mathcal{L} \subseteq \mathbb{C}^n$ is a discrete additive subgroup of $\mathbb{C}^n$ (we assume that it is a full-rank lattice). Let $\mathbf{B}$ be a matrix with as rows n linearly independent vectors in $\mathbb{C}^n$. The lattice generated by this basis is the set composed of all complex integer linear combinations of the rows of $\mathbf{B}$.*

2.2 Bounded distance decoding problem

An important problem in lattice theory is the bounded distance decoding (BDD) problem.

Definition 10 (Bounded distance decoding (BDD) problem, [DP19]). *Given a lattice $\mathcal{L} \subseteq \mathbb{R}^n$, a target vector $\boldsymbol{t} \in \mathbb{R}^n$, a unique decoding radius $r \leq \lambda_1(\mathcal{L})/2$, and the guarantee that*

$$\exists \boldsymbol{z} \in \mathcal{L} \quad s.t. \quad \|\boldsymbol{t} - \boldsymbol{z}\| < r,$$

find the lattice point $\boldsymbol{z} \in \mathcal{L}$.

Proposition 11. *The BDD problem has a unique solution due to the condition* $r \leq \lambda_1(\mathcal{L})/2$.

Proof. Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a lattice, $\boldsymbol{t} \in \mathbb{R}^n$ a target vector, and suppose

$$\exists \boldsymbol{z} \in \mathcal{L} \quad \text{s.t.} \quad \|\boldsymbol{t} - \boldsymbol{z}\| < \lambda_1(\mathcal{L})/2.$$

Assume for contradiction that there exist two lattice points $\boldsymbol{z}_1 \neq \boldsymbol{z}_2 \in \mathcal{L}$ satisfying

$$\|\boldsymbol{t} - \boldsymbol{z}_1\| < \lambda_1(\mathcal{L})/2 \quad \text{and} \quad \|\boldsymbol{t} - \boldsymbol{z}_2\| < \lambda_1(\mathcal{L})/2.$$

By the triangle inequality, we have:

$$\|\boldsymbol{z}_1 - \boldsymbol{z}_2\| = \|\boldsymbol{z}_1 - \boldsymbol{t} + \boldsymbol{t} - \boldsymbol{z}_2\| \leq \|\boldsymbol{t} - \boldsymbol{z}_1\| + \|\boldsymbol{t} - \boldsymbol{z}_2\|.$$

It then follows that $\|\boldsymbol{z}_1 - \boldsymbol{z}_2\| < \lambda_1(\mathcal{L})$. This implies that the distance between two distinct lattice points $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ is less than the minimum distance, which is a contradiction. We can conclude that the BDD problem has a unique solution. □

To provide further insight into the formal proof, we examine a simple visual example illustrated in Figure 2.1. Consider a lattice generated by the basis vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$, and let $\boldsymbol{v}_1$ be a shortest nonzero vector, such that $\lambda_1(\mathcal{L}) = \|\boldsymbol{v}_1\|$. We can then draw spheres centered at each lattice point with a radius of less than half the minimum distance, without any of the spheres overlapping. Any target point within one of these spheres is uniquely closest to the lattice point at its center. This implies that the BDD problem has a unique solution.

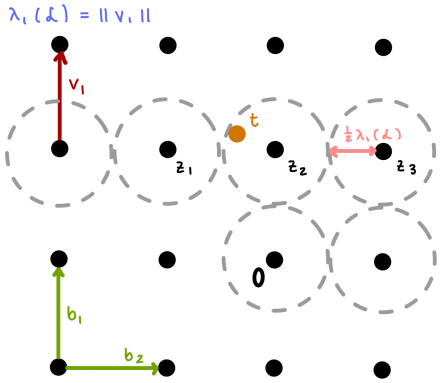


Figure 2.1: Visualization of unique solution BDD problem.

The BDD problem is also well-defined for a decoding radius larger than half the minimum distance. In this case, there may be multiple solutions and the process is called list decoding rather than bounded distance decoding [Cor20].

To determine the complexity of the algorithms used to solved the bounded distance decoding problem, the masters theorem for divide and conquer is often used. Formally, this states the following:

Theorem 12 (Masters theorem for divide and conquer recurrences, [CLRS09])**.** *The time complexity of a problem of size* n *that is solved by solving* a *subproblems of size* n/b *is given by a recurrence relation:*

$$T(n) = aT(n/b) + O\left(n^d\right)$$

where $T(n/b)$ is the time for each subproblem and $O\left(n^d\right)$ is the time to combine the solutions of the subproblems into a solution of the original problem. Then, the solution is:

$$T(n) = \begin{cases} O\left(n^d\right) & \text{if } d > \log_b a, \\ O\left(n^d \log_b n\right) & \text{if } d = \log_b a, \\ O\left(n^{\log_b a}\right) & \text{if } d < \log_b a. \end{cases}$$

Importance of the bounded distance decoding problem

The bounded distance decoding problem (BDD) plays an important role in both cryptography and communication theory.

Its computational hardness, even in quantum computing, serves as a foundation for lattice based cryptography. Specifically, unless some trapdoor information is revealed, solving the BDD problem is computationally infeasible for an adversary [LSLY].

An algorithm that solves BDD can also be used as a decoder for messages with noise in communication theory. A message is considered as a lattice point to which noise is added in transmission. Decoding this and finding the original message is an instance of the bounded distance decoding problem [DP19].

In both cryptography and communication theory, a key objective is to maximize the minimum distance between lattice points, while ensuring that the BDD problem can be solved efficiently [DP19]. In cryptography, a larger minimum distance implies a harder problem for adversaries to solve, as the larger gap between lattice points reduces the likelihood of successful guessing or approximation attacks. In communication theory, larger minimum distance increases error tolerance, meaning a larger amount of noise can be introduced to a message without compromising the ability to accurately decode the original message [Lap21].

There is a limit as to how large the minimum distance of an n-dimensional lattice in $\mathbb{R}^n$ can be. This limit is defined by Minkowski's bound.

Definition 13 (Minkowski's bound, [vW23])**.** *Let $\mathcal{L} \subseteq \mathbb{R}^n$ with basis $\mathbf{B} \in \mathbb{R}^{n \times n}$ be an n-dimensional lattice. Then*

$$\min_{\boldsymbol{x} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\boldsymbol{x}\| = \lambda_1(\mathcal{L}) \leq \sqrt{n} \cdot |\det(\mathbf{B})|^{1/n} \quad \text{or} \quad \frac{\lambda_1(\mathcal{L})}{|\det(\mathbf{B})|^{1/n}} \leq \sqrt{n}. \tag{2.1}$$

Minkowski's bound states that the normalized minimum distance of an n-dimensional lattice is less or equal than $\sqrt{n}$. One hopes to find lattices with normalized minimum distance as close as possible to this bound, while still being able to solve the bounded distance decoding problem efficiently. Unfortunately, efficient decoding algorithms are not known for lattices that are close to Minkowski's bound [DP19].

There are however efficient algorithms known for lattices with better normalized minimum distance than the baseline case, the integer lattice. An example is the efficient decoding algorithm for Barnes-Wall lattices by Micciancio and Nicolosi [MN08], which we will discuss in Section 4.1.

Chapter 3

Lattice families of interest

3.1 Barnes-Wall lattices

Barnes-Wall (BW) lattices are best described as m-dimensional lattices over the Gaussian integers $\mathbb{G} = \mathbb{Z} + i\mathbb{Z}$. Alternatively, they can be interpreted as $2m$-dimensional lattices over the integers $\mathbb{Z}$.

For a detailed justification of this equivalence, see Appendix A.1. Working with the complex representation often simplifies our definitions and algorithms.

Let k be a nonnegative integer ($k \geq 0$), and define $\ell = k+1$. Define the dimension of the BW lattice over the Gaussian integers $\mathbb{G}$ as $m = 2^k$. Consequently, define $n = 2m = 2^{k+1} = 2^\ell$, which represents the dimension of the BW lattice over the integers $\mathbb{Z}$.

Formally, we have the following definition for the BW lattices:

Definition 14 (BW lattices, [MN08])**.** *Let $k \geq 0$ be a nonnegative integer and $m = 2^k$. The k-th Barnes-Wall lattice BW_k over the Gaussian integers $\mathbb{G}$ of dimension m is the lattice generated by the rows of the k-fold Kronecker product*

$$\mathbf{B}_m = \begin{pmatrix} 1 & 1 \\ 0 & \phi \end{pmatrix}^{\otimes k} = \begin{pmatrix} \mathbf{B}_{m/2} & \mathbf{B}_{m/2} \\ \mathbf{0} & \phi\mathbf{B}_{m/2} \end{pmatrix},$$

where $\phi = 1 + i$ is the prime of the least squared norm in $\mathbb{G}$ and $\boldsymbol{B}_1 = \big(1\big)$.

Equivalently,

$$BW_{k+1} = \big\{\big(\boldsymbol{u}, \boldsymbol{u} + \phi\boldsymbol{v}\big) : \boldsymbol{u}, \boldsymbol{v} \in BW_k\big\}, \tag{3.1}$$

and $BW_0 = \mathbb{G}$.

We now discuss two properties needed to determine the normalized minimum distance of the BW lattice: the determinant and the minimum distance.

3.1.1 Determinant

Proposition 15. *Let $k \geq 0$ and let $m = 2^k$. Let BW_k be the m-dimensional BW lattice over the Gaussian integers generated by the basis $\mathbf{B}_m$. The determinant is given by:*

$$\det_{\mathbb{G}}\big(BW_k\big) = m^{m/4}. \tag{3.2}$$

Proof. We prove the correctness of formula (3.2) by using induction on k.

Base case:
Let $k = 0$ (and $m = 1$). Using the definition of the BW lattice, we have:

$$\det_{\mathbb{G}}(BW_0) = |\det_{\mathbb{G}}(\mathbf{B}_1)| = 1.$$

Since $1^{1/4} = 1$, we can conclude that the formula (3.2) holds for $k = 0$.

Inductive step:
Assume that there exists some $k \geq 0$ for which $\det_{\mathbb{G}}(BW_k) = m^{m/4}$ (inductive hypothesis). We want to show that $\det_{\mathbb{G}}(BW_{k+1}) = (2m)^{m/2}$ (formula (3.2) holds for $k+1$). We can rewrite the determinant of BW_{k+1} as follows:

$$\begin{aligned}
\det_{\mathbb{G}}(BW_{k+1}) = \det_{\mathbb{G}}(\mathcal{L}(\mathbf{B}_{2m})) &= |\det_{\mathbb{G}}(\mathbf{B}_{2m})| \\
&= |\det_{\mathbb{G}}(\mathbf{B}_m) \cdot \det_{\mathbb{G}}(\phi \cdot \mathbf{B}_m)| && (3.3) \\
&= |\det_{\mathbb{G}}(\mathbf{B}_m) \cdot \phi^m \cdot \det_{\mathbb{G}}(\mathbf{B}_n)| && (3.4) \\
&= |\phi|^m \cdot |\det_{\mathbb{G}}(\mathbf{B}_m)|^2 \\
&= \left(\sqrt{2}\right)^m \cdot (\det_{\mathbb{G}}(BW_k))^2.
\end{aligned}$$

The factorization in (3.3) uses the upper triangular structure of $\mathbf{B}_{2m}$ and (3.4) applies the property $\det(c \cdot A) = c^p \cdot \det(A)$, where c is a constant and A is a $p \times p$ matrix. By the induction hypothesis, $\det_{\mathbb{G}}(BW_k) = m^{m/4}$ and substituting this gives:

$$\det_{\mathbb{G}}(BW_{k+1}) = 2^{m/2} \cdot (\det_{\mathbb{G}}(BW_k))^2 = 2^{m/2} \cdot \left(m^{m/4}\right)^2 = 2^{m/2} \cdot m^{m/2} = (2m)^{m/2}.$$

We conclude that formula (3.2) holds for $k+1$.

This completes the induction, proving that $\det_{\mathbb{G}}(BW_k) = m^{m/4}$ for all $k \geq 0$. □

We have established the determinant of the m-dimensional BW lattice over the Gaussian integers:

$$\det_{\mathbb{G}}(BW_k) = m^{m/4}.$$

Using the isomorphism described in Appendix A.1, the corresponding real representation is an n-dimensional lattice over $\mathbb{Z}$ with $n = 2m$. The determinant of the n-dimensional BW lattice over the integers is:

$$\det_{\mathbb{Z}}(BW_k) = |\det_{\mathbb{G}}(BW_k)|^2 = \left(m^{m/4}\right)^2 = m^{m/2} = (n/2)^{n/4}.$$

3.1.2 Minimum distance

Since the minimum distance remains the same over both the complex and real numbers, we will not use a subscript to distinguish between them, as we did in the determinant calculations.

Proposition 16. *Let $k \geq 0$ and $m = 2^k$. Let BW_k be the m-dimensional BW lattice over the Gaussian integers generated by the basis $\mathbf{B}_m$. The minimum distance is:*

$$\lambda_1(BW_k) = \sqrt{2^k} = \sqrt{m}. \tag{3.5}$$

Proof. We prove the correctness of formula (3.5) by using induction on k.

Base case:
Let $k = 0$. The one-dimensional BW lattice is $\mathbb{G}$. Let $a + bi \in \mathbb{G} \setminus \{0\}$. Then:

$$\lambda_1(BW_0) = \lambda_1(\mathbb{G}) = \min_{a+bi \in \mathbb{G}\setminus\{0\}} \|a + bi\| = \min_{(a,b)\neq(0,0)} \sqrt{a^2 + b^2}.$$

We want to minimize a^2+b^2 for integers a and b with $(a, b) \neq (0, 0)$. If $(a = 1, b = 0)$, then $a^2+b^2 = 1$, if $(a = 0, b = 1)$, then $a^2 + b^2 = 1$ and if $(a = 1, b = 1)$ then $a^2 + b^2 = 2$. Hence, the smallest value of $a^2 + b^2$ that satisfies $a + bi \neq 0$ is 1 and there exist a and b that obtain this value. We conclude

$$\lambda_1(BW_0) = \min_{(a,b)\neq(0,0)} \sqrt{a^2 + b^2} = 1.$$

Since $1 = \sqrt{2^0}$, $\lambda_1(BW_k) = \sqrt{2^k}$ for $k = 0$. So formula (3.5) holds for $k = 0$.

Inductive step:
Assume that $\lambda_1(BW_k) = \sqrt{2^k}$ for some arbitrary $k \geq 0$ (inductive hypothesis). We want to show that $\lambda_1(BW_{k+1}) = \sqrt{2^{k+1}}$. Let $\boldsymbol{z} \in BW_{k+1} \setminus \{\mathbf{0}\}$. Using (3.1) we know that:

$$\boldsymbol{z} = (\boldsymbol{u}, \boldsymbol{w}) = (\boldsymbol{u}, \boldsymbol{u} + \phi\boldsymbol{v}) \tag{3.6}$$

with $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in BW_k$. Recall that the minimum distance is the length of the shortest nonzero vector. First we will prove that every nonzero vector in BW_{k+1} has length at least $\sqrt{2^{k+1}}$ and then we will prove that there exists a nonzero vector with this minimum length. We have three options for nonzero vectors.

1. Assume $\boldsymbol{u} = \mathbf{0}$ and $\boldsymbol{w} \neq \mathbf{0}$. Then $\boldsymbol{w} = \phi\boldsymbol{v}$ and

   $$\|\boldsymbol{z}\|^2 = \|\boldsymbol{u}\|^2 + \|\boldsymbol{w}\|^2 = \|\boldsymbol{w}\|^2 = \|\phi\boldsymbol{v}\|^2 = |\phi|^2 \|\boldsymbol{v}\|^2 = 2\|\boldsymbol{v}\|^2.$$

   Since $\boldsymbol{w} \neq \mathbf{0}$ it follows that $\boldsymbol{v} \neq \mathbf{0}$ and since $\boldsymbol{v} \in BW_k$, we know that $\boldsymbol{v} \in BW_k \setminus \{\mathbf{0}\}$. Then by the inductive hypothesis $2\|\boldsymbol{v}\|^2 \geq 2 \cdot 2^k = 2^{k+1}$.

2. Assume $\boldsymbol{u} \neq \mathbf{0}$ and $\boldsymbol{w} = \mathbf{0}$. Since $\boldsymbol{w} = \mathbf{0}$, it follows that $\boldsymbol{u} + \phi\boldsymbol{v} = \mathbf{0}$, so $\boldsymbol{u} = -\phi\boldsymbol{v}$ and

   $$\|\boldsymbol{z}\|^2 = \|\boldsymbol{u}\|^2 + \|\boldsymbol{w}\|^2 = \|\boldsymbol{u}\|^2 = \|-\phi\boldsymbol{v}\|^2 = |-\phi|^2 \|\boldsymbol{v}\|^2 = 2\|\boldsymbol{v}\|^2.$$

   Since $\boldsymbol{u} \neq \mathbf{0}$ it follows that $\boldsymbol{v} \neq \mathbf{0}$ and since $\boldsymbol{v} \in BW_k$, we know that $\boldsymbol{v} \in BW_k \setminus \{\mathbf{0}\}$. Then by the inductive hypothesis $\|\boldsymbol{z}\|^2 = 2\|\boldsymbol{v}\|^2 \geq 2 \cdot 2^k = 2^{k+1}$.

3. Assume $\boldsymbol{u} \neq \mathbf{0}$ and $\boldsymbol{w} \neq \mathbf{0}$. Then $\boldsymbol{u}, \boldsymbol{w} \in BW_k \setminus \{\mathbf{0}\}$. Using the inductive hypothesis, we have:

   $$\|\boldsymbol{z}\|^2 = \|\boldsymbol{u}\|^2 + \|\boldsymbol{w}\|^2 \geq 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}.$$

We can conclude that $\|\boldsymbol{z}\| \geq \sqrt{2^{k+1}}$ for all $\boldsymbol{z} \in BW_{k+1} \setminus \{\mathbf{0}\}$. Next we need to show that there exists a vector in $BW_{k+1} \setminus \{\mathbf{0}\}$ with length $\sqrt{2^{k+1}}$. By the inductive hypothesis there is some $\boldsymbol{y} \in BW_k \setminus \{\mathbf{0}\}$ with $\|\boldsymbol{y}\| = \sqrt{2^k}$. Let

$$\boldsymbol{x} = (\mathbf{0}, \phi\boldsymbol{y}) \in BW_{k+1}.$$

Then $\boldsymbol{x} \in BW_{k+1} \setminus \{\mathbf{0}\}$ and

$$\|\boldsymbol{x}\|^2 = \|\phi\boldsymbol{y}\|^2 = 2\|\boldsymbol{y}\|^2 = 2 \cdot 2^k = 2^{k+1}.$$

We conclude that every nonzero vector in BW_{k+1} has length at least $\sqrt{2^{k+1}}$ and there exists a nonzero vector in BW_{k+1} with length exactly $\sqrt{2^{k+1}}$. This completes the inductive step and thus, by induction on k, we have $\lambda_1(BW_k) = \sqrt{2^k}$ for all $k \geq 0$. □

3.2 Tensor hexagonal lattices

Let k be a nonnegative integer ($k \geq 0$), and define $m = 2^k$ and $\ell = k + 1$. Define the dimension of the tensor hexagonal lattice over the integers $\mathbb{Z}$ as $n = 2m = 2^{k+1} = 2^\ell$.

Definition 17 (Tensor hexagonal (TH) lattice)**.** *Let $\ell \geq 1$ be a positive integer and define $n = 2^\ell$. The ℓ-th tensor hexagonal lattice TH_ℓ of dimension n is the lattice over the integers generated by the rows of the ℓ-fold Kronecker product*

$$\mathbf{H}_n = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \psi \end{pmatrix}^{\otimes \ell} = \begin{pmatrix} \mathbf{H}_{n/2} & \mathbf{0} \\ \frac{1}{2}\mathbf{H}_{n/2} & \psi \mathbf{H}_{n/2} \end{pmatrix}$$

where $\psi = \frac{1}{2}\sqrt{3}$ and

$$\mathbf{H}_2 = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \psi \end{pmatrix}.$$

Equivalently,

$$TH_{\ell+1} = \left\{ \left(\boldsymbol{u} + \tfrac{1}{2}\boldsymbol{v},\ \psi \boldsymbol{v} \right) : \boldsymbol{u}, \boldsymbol{v} \in TH_\ell \right\}, \tag{3.7}$$

and $TH_1 = \mathcal{L}(\mathbf{H}_2)$.

Claim 1. *If $(\boldsymbol{x}, \boldsymbol{y}) \in TH_{\ell+1}$, then $\boldsymbol{x} \in \frac{1}{2}TH_\ell$ and $\boldsymbol{y} \in \psi TH_\ell$.*

Proof. Let $(\boldsymbol{x}, \boldsymbol{y}) \in TH_{\ell+1}$. With (3.7) we know that there exist $\boldsymbol{u}, \boldsymbol{v} \in TH_\ell$ such that:

$$\boldsymbol{x} = \boldsymbol{u} + \tfrac{1}{2}\boldsymbol{v}, \quad \boldsymbol{y} = \psi \boldsymbol{v}.$$

Since $\boldsymbol{v} \in TH_\ell$ it follows that $\frac{1}{2}\boldsymbol{v} \in \frac{1}{2}TH_\ell$ and $\psi \boldsymbol{v} \in \psi TH_\ell$. Since lattices are additive groups, the sum of elements in TH_ℓ and $\frac{1}{2}TH_\ell$ remains in the scaled lattice $\frac{1}{2}TH_\ell$. We can conclude that $\boldsymbol{x} \in \frac{1}{2}TH_\ell$ and $\boldsymbol{y} \in \psi TH_\ell$. □

3.2.1 Determinant

Proposition 18. *Let $\ell \geq 1$ and let $n = 2^\ell$. Let TH_ℓ be the n-dimensional tensor hexagonal lattice over the integers generated by the basis $\mathbf{H}_n$. The determinant of the lattice is*

$$\det(TH_\ell) = \psi^{2^{\ell-1}\ell}. \tag{3.8}$$

Proof. We prove the correctness of formula (3.8) by using induction on ℓ.

Base case:
Let $\ell = 1$ (and $n = 2$). With the definition of the tensor hexagonal lattice we find that

$$\det(TH_1) = |\det(\mathbf{H}_2)| = \psi.$$

Since $\psi^{2^{1-1}\cdot 1} = \psi^{2^0} = \psi$, we can conclude that formula (3.8) holds for $\ell = 1$.

Inductive step:
Assume that there exists some $\ell \geq 1$ for which $\det(TH_\ell) = \psi^{2^{\ell-1}\ell}$ (inductive hypothesis). We want

to show that $\det(TH_{\ell+1}) = \psi^{2^\ell(\ell+1)}$ (formula (3.8) holds for $\ell+1$). We can rewrite the determinant of $TH_{\ell+1}$ as follows:

$$\begin{aligned}
\det(TH_{\ell+1}) = \det(\mathcal{L}(\mathbf{H}_{2n})) &= |\det(\mathbf{H}_{2n})| \\
&= |\det(\mathbf{H}_n) \cdot \det(\psi \cdot \mathbf{H}_n)| && (3.9)\\
&= |\det(\mathbf{H}_n) \cdot \psi^m \cdot \det(\mathbf{H}_n)| && (3.10)\\
&= |\psi|^n \cdot |\det(\mathbf{H}_n)|^2 \\
&= \psi^n \cdot (\det(TH_\ell))^2 .
\end{aligned}$$

The factorization in (3.9) uses the bottom triangular structure of $\mathbf{H}_{2n}$ and (3.10) applies the property $\det(c \cdot A) = c^p \cdot \det(A)$, where c is a constant and A is square matrix of dimension p. By the inductive hypothesis, $\det(TH_\ell) = \psi^{2^{\ell-1}\ell}$ and substituting this gives:

$$\det(TH_{\ell+1}) = \psi^{2^\ell} \cdot (\det(TH_\ell))^2 = \psi^{2^\ell} \cdot \left(\psi^{2^{\ell-1}\ell}\right)^2 = \psi^{2^\ell + 2\cdot 2^{\ell-1}\ell} = \psi^{2^\ell + 2^\ell \ell} = \psi^{2^\ell(\ell+1)}.$$

We conclude that formula (3.8) holds for $\ell + 1$.

This completes the induction, proving that $\det(TH_\ell) = \psi^{2^{\ell-1}\ell}$ for all $\ell \geq 1$. □

We have established the determinant of the TH lattice of dimension $n = 2^\ell$ over the integers. We can also express this determinant in terms of n, this gives us the following for $\ell \geq 1$:

$$\det(TH_\ell) = \psi^{2^{\ell-1}\ell} = \psi^{n \log_2(n)/2}.$$

3.2.2 Minimum distance

Proposition 19. *Let $\ell \geq 1$ and $n = 2^\ell$. Let TH_ℓ be the n-dimensional TH lattice over the integers generated by the basis $\mathbf{H}_n$. The minimum distance is:*

$$\lambda_1(TH_\ell) = 1. \tag{3.11}$$

Proof. We prove the correctness of (3.11) by using induction on ℓ.

Base case:
Let $\ell = 1$. Let $\boldsymbol{z} \in TH_1$, then

$$\boldsymbol{z} = x_1 \cdot (1, 0) + x_2 \cdot \left(\tfrac{1}{2}, \psi\right)$$

for some $x_1, x_2 \in \mathbb{Z}$. The squared length of the vector is:

$$\|\boldsymbol{z}\|^2 = \left\|\left(x_1 + \tfrac{1}{2}x_2,\ \psi x_2\right)\right\|^2 = \left(x_1 + \tfrac{1}{2}x_2\right)^2 + (\psi x_2)^2 = x_1^2 + x_1 x_2 + x_2^2.$$

We want to minimize $x_1^2 + x_1x_2 + x_2^2$ for integers x_1 and x_2 with $(x_1, x_2) \neq (0, 0)$. If $x_1 = 0$ and $x_2 \neq 0$, we have $\|\boldsymbol{z}\|^2 = x_2^2$ which is minimal for $x_2 = 1, -1$. For $x_1 \neq 0$ and $x_2 = 0$ we have $\|\boldsymbol{z}\|^2 = x_1^2$, which is minimal for $x_1 = 1, -1$. For $x_1 \neq 0$ and $x_2 \neq 0$ all terms contribute to the minimum distance. For $x_1 = 1$ and $x_2 = 1$ we have $\|\boldsymbol{z}\|^2 = 3 \geq 1$, for $x_1 = 1$ and $x_2 = -1$ we have $\|\boldsymbol{z}\| = 1 \geq 1$, for $x_1 = -1$ and $x_2 = 1$ we have $\|\boldsymbol{z}\|^2 = 1 \geq 1$ and lastly for $x_1 = -1$ and $x_2 = -1$ we have $\|\boldsymbol{z}\|^2 = 3 \geq 1$. Any other values outside of $-1, 0$ and 1 will give larger squared lengths. We conclude that for all nonzero vectors the squared length is larger or equal to 1, which implies $\|\boldsymbol{z}\| \geq 1$ for all $\boldsymbol{z} \in TH_1 \setminus \{\mathbf{0}\}$. It is easy to see that any other values outside of $-1, 0$ and 1 will give larger squared lengths. So we can conclude that for all nonzero vectors the squared length is larger or equal to 1. We also know that there are 6 vectors in TH_1 with length 1 (use the values for x_1 and x_2).

Inductive step:
Assume that $\lambda_1(TH_\ell) = 1$ for some arbitrary $\ell \geq 1$ (inductive hypothesis). We want to show that $\lambda_1(TH_{\ell+1}) = 1$. Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ be an arbitrary vector in $TH_{\ell+1} \setminus \{\mathbf{0}\}$. Because of Definition 3.7 and Claim 1 we have

$$(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{u} + \tfrac{1}{2}\boldsymbol{v}, \psi\boldsymbol{v})$$

with $\boldsymbol{u}, \boldsymbol{v} \in TH_\ell$. Recall that the minimum distance is the length of the shortest nonzero vector. First we will prove that every nonzero vector in $TH_{\ell+1}$ has length at least 1 and then we will prove that there exists a nonzero vector with length exactly 1. We have three options for nonzero vectors.

1. Assume $\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{y} \neq \mathbf{0}$. Since $\boldsymbol{x} = \mathbf{0}$ it follows that $\boldsymbol{u} + \frac{1}{2}\boldsymbol{v} = \mathbf{0}$, so $\boldsymbol{v} = -2\boldsymbol{u}$ and then:

$$\|\boldsymbol{z}\|^2 = \|\boldsymbol{y}\|^2 = \left\|\tfrac{1}{2}\sqrt{3} \cdot \boldsymbol{v}\right\|^2 = \left\|\tfrac{1}{2}\sqrt{3} \cdot -2\boldsymbol{u}\right\|^2 = \left\|-\sqrt{3}\boldsymbol{u}\right\|^2 \geq 3.$$

2. Assume $\boldsymbol{x} \neq \mathbf{0}$ and $\boldsymbol{y} = \mathbf{0}$. Since $\boldsymbol{y} = \mathbf{0}$ it follows that $\boldsymbol{v} = \mathbf{0}$, so $\boldsymbol{x} = \boldsymbol{u}$. Since $\boldsymbol{x} \neq \mathbf{0}$ we know that $\boldsymbol{u} \in TH_k \setminus \{\mathbf{0}\}$. With the inductive hypothesis we know that $\|\boldsymbol{u}\| \geq 1$, hence:

$$\|\boldsymbol{z}\|^2 = \|\boldsymbol{x}\|^2 = \|\boldsymbol{u}\|^2 \geq 1.$$

3. Assume $\boldsymbol{x} \neq \mathbf{0}$ and $\boldsymbol{y} \neq \mathbf{0}$. Using Claim 1, we know that $\boldsymbol{x} \in \frac{1}{2}TH_\ell$ and $\boldsymbol{y} \in \psi TH_\ell$, so there exists $\boldsymbol{w} \in \frac{1}{2}TH_k \setminus \{\mathbf{0}\}$ and $\boldsymbol{v} \in \psi TH_k \setminus \{\mathbf{0}\}$ such that $(\boldsymbol{x}, \boldsymbol{y}) = (\frac{1}{2}\boldsymbol{w}, \psi\boldsymbol{v})$. Using the inductive hypothesis we then find the following:

$$\|\boldsymbol{z}\|^2 = \left\|\left(\tfrac{1}{2}\boldsymbol{w}, \psi\boldsymbol{v}\right)\right\|^2 = \left\|\tfrac{1}{2}\boldsymbol{w}\right\|^2 + \|\psi\boldsymbol{v}\|^2 = \tfrac{1}{4}\|\boldsymbol{w}\|^2 + \tfrac{3}{4}\|\boldsymbol{v}\|^2 \geq 1.$$

We can conclude that $\|\boldsymbol{z}\|^2 \geq 1$ for all $\boldsymbol{z} \in TH_\ell \setminus \{\mathbf{0}\}$. Next we need to show that there exists a vector in $TH_\ell \setminus \{\mathbf{0}\}$ with length 1. By the inductive hypothesis we know that there exists a nonzero vector $\boldsymbol{u}'$ in TH_ℓ with length 1. Let $\boldsymbol{z}' = (\boldsymbol{u}', \mathbf{0}) \in TH_{\ell+1}$. Then $\boldsymbol{z}' \in TH_{\ell+1} \setminus \{0\}$ and $\|\boldsymbol{z}'\| = 1$. We conclude that every nonzero vector in $TH_{\ell+1}$ has length 1. This completes the inductive step and thus, by induction on ℓ, we have $\lambda_1(TH_\ell) = 1$ for all $\ell \geq 1$. □

3.3 Comparing lattices

Let $k \geq 0$ and $m = 2^k$. Let $\ell = k + 1$ and $n = 2^\ell = 2^{k+1} = 2m$.

To fairly compare different lattice families, we use the normalized minimum distance. As a baseline, we consider the normalized minimum distance of the n-dimensional integer lattice $\mathbb{Z}^n$, which remains constant across dimensions. In $\mathbb{Z}^n$ the minimum distance is 1, since the shortest nonzero vectors are the standard basis vectors. The determinant of $\mathbb{Z}^n$ is also 1, as it is generated by the identity matrix. Therefore, the normalized minimum distance is equal to:

$$\frac{\lambda_1(\mathbb{Z}^n)}{\det(\mathbb{Z}^n)^{1/n}} = 1.$$

The best case is Minkowski's bound, which is for an n-dimensional lattice is equal to:

$$\Theta(\sqrt{n}).$$

Let BW_k be the m-dimensional BW lattice over the Gaussian integers. The corresponding real representation is a lattice over $\mathbb{Z}$ of dimension $2m = n$. Using the determinant and minimum distance

calculated in the previous sections, we find that the normalized minimum distance over the integers is equal to:

$$\frac{\lambda_1\left(BW_k\right)}{\det_{\mathbb{Z}}\left(BW_k\right)^{1/2m}} = \frac{m^{1/2}}{\left(m^{m/2}\right)^{1/2m}} = \frac{m^{1/2}}{m^{1/4}} = m^{1/4} = (n/2)^{1/4}.$$

For large n, the normalized minimum distance scales as $n^{1/4}$ and so the growth is proportional to $n^{1/4}$. The normalized minimum distance of the n-dimensional BW lattice over the integers simplifies to

$$\Theta\left(n^{1/4}\right).$$

Let TH_ℓ be the n-dimensional tensor hexagonal lattice over the integers. Using the determinant and minimum distance calculated in previous sections, we find that the normalized minimum distance over the integers is equal to:

$$\frac{\lambda_1\left(TH_\ell\right)}{\left|\det_{\mathbb{Z}}\left(\mathbf{H}_n\right)\right|^{1/n}} = \frac{1}{\left(\psi^{n\log_2(n)/2}\right)^{1/n}} = \psi^{-\log_2(n)/2} = \left(\tfrac{4}{3}\right)^{\frac{1}{4}\log_2(n)}.$$

Using the property $a^{\log_b(c)} = c^{\log_b(a)}$ we can rewrite this expression as follows:

$$\left(\tfrac{4}{3}\right)^{\log_2(n)/4} = n^{\log_2\left((4/3)^{1/4}\right)} = n^{\log_2(4/3)/4} = n^{(\log_2(4)-\log_2(3))/4} = n^{(2-\log_2(3))/4} \approx n^{0.1038}.$$

We can conclude that the normalized minimum distance of the tensor hexagonal lattice TH_ℓ is given by:

$$\Theta\left(n^{0.1038}\right).$$

In conclusion, a good baseline for the normalized minimum distance is the integer lattice, as it doesn't grow with dimension. The tensor hexagonal lattice with $\Theta\left(n^{0.1038}\right)$ and Barnes-Wall lattice with $\Theta\left(n^{1/4}\right)$ both outperform the integer lattice but fall short of Minkowski's bound $\Theta\left(\sqrt{n}\right)$. Among these, the BW lattice has a higher growth rate of normalized minimum distance compared to the tensor hexagonal lattice.

Chapter 4

Bounded distance decoding problem

In this chapter we are going to discuss algorithms for solving the bounded distance decoding (BDD) problem in lattices. We begin with an algorithm by Micciancio and Nicolosi, which is specific to Barnes-Wall lattices [MN08]. We then deconstruct and generalize this algorithm, providing a framework to solve the BDD problem in a broader class of lattices. Lastly, we demonstrate the general algorithm's versatility by applying it to the tensor hexagonal lattice.

4.1 Solving the bounded distance decoding problem in Barnes-Wall lattices

In this section we consider Barnes-Wall lattices as lattices over the Gaussian integers. BW lattices are a family of lattices with a strong recursive structure, as we have seen in the previous chapter. This allows for efficient decoding algorithms, as given by Micciancio and Nicolosi [MN08]. We consider a version of the algorithm without parallelization and assume that we are not restricted by the number of available processors. The bounded distance decoding problem in BW lattices is formally defined as:

Definition 20 (BDD problem in BW lattices)**.** *Let $k \geq 0$ and $m = 2^k$. Let BW_k the m-dimensional BW lattice. Let $\boldsymbol{t} \in \mathbb{C}^m$ be a target vector and assume that*

$$\exists \boldsymbol{z} \in BW_k \quad s.t. \quad \|\boldsymbol{t} - \boldsymbol{z}\|^2 < m/4.$$

Find the lattice point $\boldsymbol{z} \in BW_k$.

Claim 2. *Solving this BDD instance will give a unique solution.*

Proof. The minimum distance in the m-dimensional BW lattice over the Gaussian integers is $\sqrt{m}$. The unique decoding radius, the radius for which the BDD problem has a unique solution, is half the minimum distance, i.e. $\sqrt{m}/2$. The squared unique decoding radius is $m/4$ (we work with squared distances for easier computation). Solving the BDD instance will give a unique lattice point if the squared distance between the target vector and lattice vector is less than $m/4$, which is the case. □

The algorithm given by Micciancio and Nicolosi [MN08] to solve the bounded distance decoding problem in BW lattices is based on the following four observations:

1. If $(\boldsymbol{z}_0, \boldsymbol{z}_1) \in BW_{k+1}$, then $\boldsymbol{z}_0, \boldsymbol{z}_1 \in BW_k$.

2. Assume $m \geq 2$. Let $\boldsymbol{t} = (\boldsymbol{t}_0, \boldsymbol{t}_1) \in \mathbb{C}^m$ and $\boldsymbol{z} = (\boldsymbol{z}_0, \boldsymbol{z}_1) \in BW_k$ and assume that $\boldsymbol{t}$ is within squared unique decoding radius of $\boldsymbol{z}$, so $\|\boldsymbol{t} - \boldsymbol{z}\|^2 < m/4$. Per definition we have:

$$\|\boldsymbol{t} - \boldsymbol{z}\|^2 = \|(\boldsymbol{t}_0 - \boldsymbol{z}_0, \boldsymbol{t}_1 - \boldsymbol{z}_1)\|^2 = \|\boldsymbol{t}_0 - \boldsymbol{z}_0\|^2 + \|\boldsymbol{t}_1 - \boldsymbol{z}_1\|^2 .$$

   Hence $\|\boldsymbol{t}_0 - \boldsymbol{z}_0\|^2 + \|\boldsymbol{t}_1 - \boldsymbol{z}_1\|^2 < m/4$. This is only possible if at least one of the two squared distances is less than $m/8$ (if both are larger than or equal to $m/8$, then the total sum is larger than or equal to $m/4$, contradiction). So either $\boldsymbol{t}_0$ is within the squared unique decoding radius of BW_{k-1} or $\boldsymbol{t}_1$ is within the squared unique decoding radius of BW_{k-1}, or both. We conclude that if $\boldsymbol{t}$ is within the squared unique decoding radius of BW_k, then at least one among $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ has to be within squared unique decoding radius of BW_{k-1}.

3. The function

$$\mathcal{T} : (\boldsymbol{z}_0, \boldsymbol{z}_1) \mapsto (\phi/2) \cdot (\boldsymbol{z}_0 - \boldsymbol{z}_1, \boldsymbol{z}_0 + \boldsymbol{z}_1)$$

   is an automorphism of BW_k.

   Proof. To show that $\mathcal{T}$ is an automorphism, we need to proof that $\|\boldsymbol{z}\|^2 = \|\mathcal{T}(\boldsymbol{z})\|$.

$$\begin{aligned}
\|\mathcal{T}(\boldsymbol{z})\|^2 &= \|(\phi/2) \cdot (\boldsymbol{z}_0 - \boldsymbol{z}_1, \boldsymbol{z}_0 + \boldsymbol{z}_1)\|^2 = |(\phi/2)|^2 \cdot \|(\boldsymbol{z}_0 - \boldsymbol{z}_1, \boldsymbol{z}_0 + \boldsymbol{z}_1)\|^2 \\
&= \tfrac{1}{2} \cdot \|(\boldsymbol{z}_0 - \boldsymbol{z}_1, \boldsymbol{z}_0 + \boldsymbol{z}_1)\|^2 \\
&= \tfrac{1}{2} \cdot \|\boldsymbol{z}_0 - \boldsymbol{z}_1\|^2 + \tfrac{1}{2} \cdot \|\boldsymbol{z}_0 + \boldsymbol{z}_1\|^2 = \tfrac{1}{2} \cdot (\boldsymbol{z}_0 - \boldsymbol{z}_1)^2 + \tfrac{1}{2} \cdot (\boldsymbol{z}_0 + \boldsymbol{z}_1)^2 \\
&= \tfrac{1}{2} \cdot (\boldsymbol{z}_0^2 - 2\boldsymbol{z}_0\boldsymbol{z}_1 + \boldsymbol{z}_1^2) + \frac{1}{2} \cdot (\boldsymbol{z}_0^2 + 2\boldsymbol{z}_0\boldsymbol{z}_1 + \boldsymbol{z}_1^2) \\
&= \tfrac{1}{2} \cdot (2\boldsymbol{z}_0^2 - 2\boldsymbol{z}_0\boldsymbol{z}_1 + 2\boldsymbol{z}_0\boldsymbol{z}_1 + 2\boldsymbol{z}_1^2) \\
&= \boldsymbol{z}_0^2 + \boldsymbol{z}_1^2 = \|\boldsymbol{z}_0\|^2 + \|\boldsymbol{z}_1\|^2 = \|(\boldsymbol{z}_0, \boldsymbol{z}_1)\|^2 = \|\boldsymbol{z}\|^2 .
\end{aligned}$$

   So we can conclude that the function $\mathcal{T}$ is indeed an automorphism. □

4. Let $(\boldsymbol{z}_-, \boldsymbol{z}_+) = \mathcal{T}((\boldsymbol{z}_0, \boldsymbol{z}_1))$. The vectors $\boldsymbol{z}_0$ and $\boldsymbol{z}_1$ can be recovered from any of the pairs

$$(\boldsymbol{z}_0, \boldsymbol{z}_-), (\boldsymbol{z}_0, \boldsymbol{z}_+), (\boldsymbol{z}_1, \boldsymbol{z}_-), (\boldsymbol{z}_1, \boldsymbol{z}_+) .$$

   Proof. We are going to show this for one of the pairs, but the rest can be recovered using a similar approach. If we know $\boldsymbol{z}_0$ and we want to recover $\boldsymbol{z}_1$ when knowing $\boldsymbol{z}_-$, we get the following:

$$\boldsymbol{z}_- = (\phi/2) \cdot (\boldsymbol{z}_0 - \boldsymbol{z}_1) = \boldsymbol{z}_- = (\phi/2) \cdot \boldsymbol{z}_0 - (\phi/2) \cdot \boldsymbol{z}_1 \Leftrightarrow 2\phi^{-1}\boldsymbol{z}_- = \boldsymbol{z}_0 - \boldsymbol{z}_1 \Leftrightarrow \boldsymbol{z}_1 = \boldsymbol{z}_0 - 2\phi^{-1}\boldsymbol{z}_- .$$

   This is how we can recover the original from two of the projections. □

The algorithm given by Micciancio and Nicolosi is based on these observations. It works by recursively (and independently) decoding 4 target vectors of dimension $m/2 = 2^{k-1}$ that are derived from the received target vector and then combining the results appropriately [GP12]. The main feature of the algorithm is the use of a distance-preserving linear automorphism $\mathcal{T}$ of the BW lattice.

Algorithm 1 Decoding algorithm for Barnes-Wall lattices of dimension $m = 2^k$

function DECODEBW($\boldsymbol{t}$)
  if $\boldsymbol{t} \in \mathbb{C}^1$ **then**
    return $\lceil \boldsymbol{t} \rfloor$ — Round $\boldsymbol{t}$ component-wise to the closest Gaussian integer
  else
    $(\boldsymbol{t}_0, \boldsymbol{t}_1) \leftarrow \boldsymbol{t}$ — Split $\boldsymbol{t}$ into two halves
    $(\boldsymbol{t}_+, \boldsymbol{t}_-) \leftarrow (\phi/2) \cdot (\boldsymbol{t}_0 + \boldsymbol{t}_1, \boldsymbol{t}_0 - \boldsymbol{t}_1)$ — Compute $\mathcal{T}(\boldsymbol{s})$
    for all $i \in \{0, 1, +, -\}$ **do**
      $\boldsymbol{z}_i \leftarrow$ DECODEBW($\boldsymbol{t}_i$) — Execute the recursive calls on $\boldsymbol{t}_0, \boldsymbol{t}_1, \mathcal{T}(\boldsymbol{t}_0), \mathcal{T}(\boldsymbol{t}_1)$
    end for
    $\boldsymbol{z}_0^+ \leftarrow (\boldsymbol{z}_0, 2\phi^{-1}\boldsymbol{z}_+ - \boldsymbol{z}_0)$ — Compute the 4 candidate vectors $(\boldsymbol{z}_0, \boldsymbol{z}_1) \in BW_k$
    $\boldsymbol{z}_0^- \leftarrow (\boldsymbol{z}_0, \boldsymbol{z}_0 - 2\phi^{-1}\boldsymbol{z}_-)$
    $\boldsymbol{z}_1^+ \leftarrow (2\phi^{-1}\boldsymbol{z}_+ - \boldsymbol{z}_1, \boldsymbol{z}_1)$
    $\boldsymbol{z}_1^- \leftarrow (2\phi^{-1}\boldsymbol{z}_- + \boldsymbol{z}_1, \boldsymbol{z}_1)$
    $\boldsymbol{z} \leftarrow \arg\min_{\boldsymbol{z}' \in \{\boldsymbol{z}_0^+, \boldsymbol{z}_0^-, \boldsymbol{z}_1^+, \boldsymbol{z}_1^-\}} \{\|\boldsymbol{t} - \boldsymbol{z}'\|\}$ — Select the candidate closest to $\boldsymbol{t}$
    return $\boldsymbol{z}$
  end if
end function

Next we are going to proof the correctness of this algorithm and determine the complexity.

4.1.1 Correctness

Theorem 21. *For any $m = 2^k$ and $\boldsymbol{t} \in \mathbb{C}^m$ such that the squared distance between $\boldsymbol{t}$ and the m-dimensional BW lattice is less than $m/4$, Algorithm 1 computes the unique lattice vector $\boldsymbol{z} \in BW_k$ within squared distance $m/4$ from the target vector $\boldsymbol{t}$ [MN08].*

Proof. The base case is clear, so we only need to check correctness for $m \geq 2$. Let $(\boldsymbol{z}_0, \boldsymbol{z}_1)$ be the lattice point within squared distance $m/4$ from the target $(\boldsymbol{t}_0, \boldsymbol{t}_1)$. Since $\mathcal{T}$ is an automorphism, the lattice point $(\boldsymbol{z}_-, \boldsymbol{z}_+) = \mathcal{T}((\boldsymbol{z}_0, \boldsymbol{z}_1))$ is within squared distance $m/4$ from the target $(\boldsymbol{t}_-, \boldsymbol{t}_+) = \mathcal{T}((\boldsymbol{t}_0, \boldsymbol{t}_1))$. With the previously discussed observations, we know that at least one among $\boldsymbol{t}_0$ or $\boldsymbol{t}_1$ is within the squared unique decoding radius in BW_{k-1}, and that at least one among $\mathcal{T}(\boldsymbol{t}_0)$ or $\mathcal{T}(\boldsymbol{t}_1)$ is within the squared unique decoding radius in BW_{k-1}.

The algorithm recursively computes four $m/2$-dimensional vectors, and we have four potential pairings of these vectors where one of them is guaranteed to indeed be the closest lattice vector in dimension m, namely $(\boldsymbol{z}_0, \boldsymbol{z}_-)$, $(\boldsymbol{z}_0, \boldsymbol{z}_+)$, $(\boldsymbol{z}_1, \boldsymbol{z}_-)$ and $(\boldsymbol{z}_1, \boldsymbol{z}_+)$. From the previous observations we know that the original lattice vector can be recovered from any of these pairings and we know that there is only one vector within the squared unique decoding radius. Hence, recovering the original lattice vector and selecting the candidate vector closest to the target, will give you the correct solution for the BDD problem in the m-dimensional BW lattice. So, Algorithm 1 computes the unique lattice vector $\boldsymbol{z} \in BW_k$ within squared distance $m/4$ from the target vector $\boldsymbol{t}$. □

4.1.2 Complexity

Theorem 22. *For any $m = 2^k$ and $\boldsymbol{t} \in \mathbb{C}^m$, the execution of Algorithm 1 terminates after $O(m^2)$ steps.*

Proof. At each recursion depth the problem is divided into four subproblems of size $m/2$ and after obtaining the solutions to these subproblems, the algorithm combines them and selects the best ones. The combining and selecting has a complexity of $O(m)$ (it is a linear amount of work). Let $T(m)$ denote the runtime for decoding the BW lattice of dimension m. We can then conclude the following:

$$T(m) = 4T(m/2) + O(m).$$

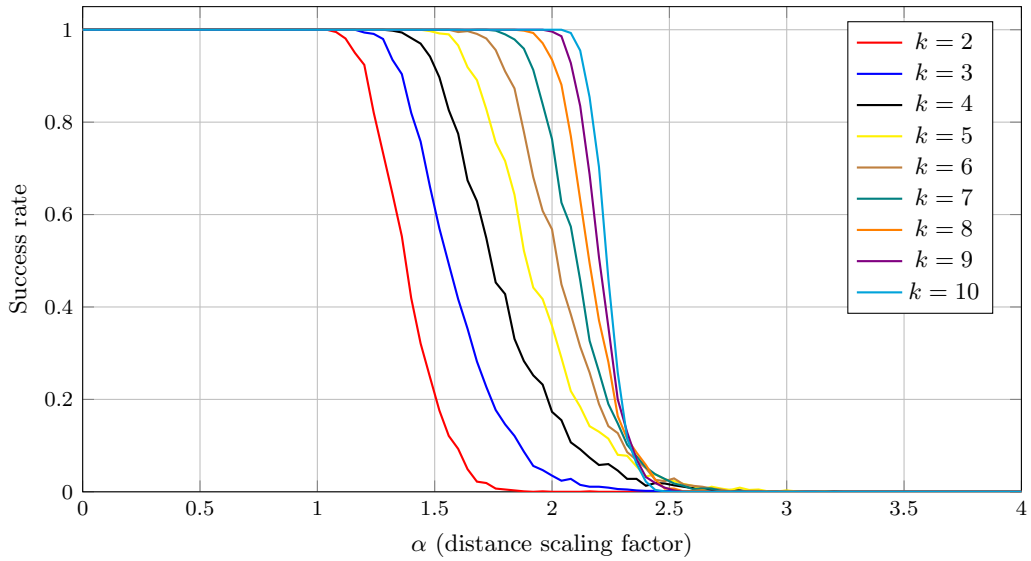
Using the masters theorem for divide and conquer recurrences, as given in Theorem 12, we can solve this recurrence relation. To apply the theorem we use $a = 4, b = 2, d = 1$. It then follows that $\log_b a = 2$ and $d = 1 < 2$, so $T(m) = O\left(m^{\log_b a}\right) = O\left(m^2\right)$. So we can conclude that the algorithm terminates after $O\left(m^2\right)$ steps. □

4.1.3 Performance outside the squared unique decoding radius

In Theorem 21 we proved that Algorithm 1 computes the unique lattice vector $\boldsymbol{z} \in BW_k$ within squared distance $m/4$ from some target vector $\boldsymbol{t}$. The squared unique decoding radius represents a theoretical boundary within which the decoder is guaranteed to succeed. To test the robustness of the algorithm, we can test how the algorithm performs for values outside the squared unique decoding radius.

To investigate this, we implemented the algorithm in Python. The complete implementation, including the tests that were run, can be found in Appendix A.2. We generated target vectors $\boldsymbol{t}$ at varying distances of $\alpha \cdot (m/4)$ from lattice points in the Barnes-Wall lattice and evaluated if the algorithm returned the correct lattice point. We tested for dimensions 2^k with $k \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. The experiments were run on 32 processors, 1000 samples per datapoint, and 100 datapoints per curve.

Decoding performance within squared distance $\alpha \cdot (m/4)$ from a certain lattice point



The results show us that when the algorithm operates outside the unique decoding radius, the success rate of correctly identifying the lattice point decreases. We also observe that higher-dimensional lattices maintain a higher decoding success rate for longer outside the unique decoding radius compared to lower-dimensional lattices.

To understand why higher-dimensional BW lattices sustain higher success rates longer than lower-dimensional BW lattices, we consider open balls of radius $\alpha \cdot (m/4)$ around each lattice point in the BW lattice. In low dimensions, as the radius increases, the open balls around adjacent lattice points begin to overlap quickly. This rapid overlap means that a point quickly is within the radius of multiple lattice points, causing a decrease in the success rate of the algorithm. In higher dimensions, the empty space between the open balls increases exponentially. This greater separation delays significant overlaps, which causes an extended plateau in the succes rate.

4.2 Solving the bounded distance decoding problem in lattices of dimension $n = 2^\ell$, $\ell \in \mathbb{Z}_{\geq 1}$

In the previous section, we discussed an algorithm for solving the bounded distance decoding problem in the Barnes-Wall lattice. The key idea of this algorithm is to project the lattice onto smaller dimensions, solve the decoding problem in these reduced dimensions, and then reconstruct the original lattice vector using the solutions from the reduced dimensions. This general idea can be used to formulate an algorithm for solving the BDD problem in lattices other than the Barnes-Wall lattice.

4.2.1 General decoding algorithm

Let $\ell \in \mathbb{Z}_{\geq 1}$ and $n = 2^\ell$. Let $\mathcal{L}$ be an n-dimensional lattice over the real numbers. Using the ideas from the decoding algorithm for BW lattices we will now formulate a generalized algorithm for solving the bounded distance decoding problem in $\mathcal{L}$.

The correctness of the algorithm depends on a series of hypotheses. These hypotheses describe properties of the lattice and projections that must hold to ensure that the algorithm will find the correct lattice point.

Hypothesis 1. *There exist s projections $\pi_1, \ldots, \pi_s : \mathbb{R}^n \to \mathbb{R}^{n/2}$ such that*

$$\pi_i(\mathcal{L}) \simeq \mathcal{L}' \quad \forall i \in \{1, \ldots, s\}$$

for some lattice $\mathcal{L}'$ with $\dim(\mathcal{L}') = n/2$.

Hypothesis 2. *For any two projections π_i and π_j with $i \neq j \leq s$ it holds that:*

$$\ker(\pi_i) \cap \ker(\pi_j) = \{\mathbf{0}\}.$$

Hypothesis 3. *For any $\boldsymbol{e} \in \mathbb{R}^n$, there exist $i \neq j \leq s$ and $\gamma < 1$ such that:*

$$\|\pi_i(\boldsymbol{e})\| \leq \gamma \cdot \|\boldsymbol{e}\| \quad \text{and} \quad \|\pi_j(\boldsymbol{e})\| \leq \gamma \cdot \|\boldsymbol{e}\|.$$

Hypothesis 4. *There exists an algorithm that solves the bounded distance decoding problem in the lattice $\mathcal{L}'$ up to an absolute radius $\gamma \cdot r$ with $\gamma \cdot r \leq \lambda_1(\mathcal{L}')/2$. This algorithm is called the subroutine in the decoding algorithm.*

These four hypotheses ensure that the algorithm solves the bounded distance decoding problem correctly. We will formally proof this in the next section.

Algorithm 2 General decoding algorithm for lattices of dimension $n = 2^\ell$, $\ell \in \mathbb{Z}_{\geq 1}$

1 **function** DECODE($\boldsymbol{t}$)
2 **if** $\boldsymbol{t} \in \mathbb{R}^1$ **then**
3 **return** $\lceil \boldsymbol{t} \rfloor$ — Round $\boldsymbol{t}$ to the closest integer
4 **else**
5 $m \leftarrow \infty$
6 **for all** $i \in \{1, \ldots, s\}$ **do**
7 $\boldsymbol{t}_i \leftarrow \pi_i(\boldsymbol{t})$ — Compute the projection of the target vector
8 $\boldsymbol{z}_i \leftarrow$ SUBROUTINE$(\pi_i(\boldsymbol{t}))$ — Decode in the projected lattice
9 **end for**
10 **for all** $i, j \in \{1, \ldots, s\}$ with $i < j$ **do**
11 $\boldsymbol{z}_j^i \leftarrow$ RECONSTRUCT$(\boldsymbol{z}_i, \boldsymbol{z}_j)$ — Reconstruct original from any two projections
12 **if** $\|\boldsymbol{t} - \boldsymbol{z}_j^i\| < m$ **then** — Store the vector closest to the target vector
13 $m \leftarrow \|\boldsymbol{t} - \boldsymbol{z}_j^i\|$, $\boldsymbol{z} \leftarrow \boldsymbol{z}_j^i$
14 **end if**
15 **end for**
16 **return** $\boldsymbol{z}$
17 **end if**
18 **end function**

4.2.2 Correctness

Theorem 23. *Let $\mathcal{L}$ be a lattice of dimension $n = 2^\ell$ with $\ell \in \mathbb{Z}_{\geq 1}$. For any $\boldsymbol{t} \in \mathbb{R}^n$ such that the distance between $\boldsymbol{t}$ and $\mathcal{L}$ is less than half the minimum distance $r = \lambda_1(\mathcal{L})/2$ and the stated hypotheses are satisfied, Algorithm 2 computes the unique lattice vector $\boldsymbol{z} \in \mathcal{L}$ within distance r from the target vector $\boldsymbol{t}$.*

Proof. The first step in the algorithm is to compute the projections of the target vector and then decode these projections in the projected lattice. Using Hypothesis 3, there exist indices $i \neq j \leq s$ such that

$$\|\pi_i(\boldsymbol{e})\| \leq \gamma \cdot \|\boldsymbol{e}\| \quad \text{and} \quad \|\pi_j(\boldsymbol{e})\| \leq \gamma \cdot \|\boldsymbol{e}\|,$$

where $\boldsymbol{e} = \boldsymbol{t} - \boldsymbol{z}$. Since $\|\boldsymbol{e}\| < r$ (definition of the bounded distance decoding problem) and $\gamma < 1$, it follows that:

$$\|\pi_i(\boldsymbol{e})\| < \gamma \cdot r \quad \text{and} \quad \|\pi_j(\boldsymbol{e})\| < \gamma \cdot r.$$

By Hypothesis 1, we know that the projections are isomorphic to a lattice $\mathcal{L}'$ and by Hypothesis 4, there exists a subroutine that can decode correctly up to an absolute radius $\gamma \cdot r$ in $\mathcal{L}$. As a result, the algorithm can correctly recover the lattice vector $\boldsymbol{z}_i \in \pi_i(\mathcal{L})$ from $\boldsymbol{t}_i = \pi_i(\boldsymbol{t})$ whenever $\|\pi_i(\boldsymbol{e})\| < \gamma \cdot r$. Therefore, the algorithm correctly decodes the projections $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ in line 8 of the algorithm.

After decoding the projections, the next step is to reconstruct the original lattice vector $\boldsymbol{z} \in \mathcal{L}$ from the decoded projections $\boldsymbol{z}_i \in \pi_i(\mathcal{L})$ and $\boldsymbol{z}_j \in \pi_j(\mathcal{L})$. Hypothesis 2 ensures that for any two distinct projections π_i and π_j, we have:

$$\ker(\pi_i) \cap \ker(\pi_j) = \{\boldsymbol{0}\}.$$

Equivalently, the linear map $\boldsymbol{z} \mapsto (\pi_i(\boldsymbol{z}), \pi_j(\boldsymbol{z}))$ is injective. This implies that no nonzero vector is sent to $(\boldsymbol{0}, \boldsymbol{0})$ by both projections. It follows that any vector $\boldsymbol{z} \in \mathcal{L}$ can be uniquely determined from its images $\pi_i(\boldsymbol{z})$ and $\pi_j(\boldsymbol{z})$.

More concretely, let A_i and A_j denote the matrices representing the projections π_i and π_j. Once we have decoded $\boldsymbol{z}_i = \pi_i(\boldsymbol{z})$ and $\boldsymbol{z}_j = \pi_j(\boldsymbol{z})$, the original vector $\boldsymbol{z}$ can be reconstructed by solving

$$\begin{pmatrix} A_i \\ A_j \end{pmatrix} \boldsymbol{z} = \begin{pmatrix} \boldsymbol{z}_i \\ \boldsymbol{z}_j \end{pmatrix}.$$

Since $\ker(\pi_i) \cap \ker(\pi_j) = \{\mathbf{0}\}$, this system of equations has a unique solution $\boldsymbol{z}$.

Lastly, the algorithm computes the distance between the target vector and each candidate lattice vector reconstructed from pairs of projections. By Hypothesis 2, the reconstruction is correct for any pair of projections π_i and π_j, and Hypothesis 4 ensures that the decoding radius in the projected lattices is sufficient for correct decoding. The algorithm iterates over all $\binom{s}{2}$ pairs of projections, reconstructs the corresponding candidate lattice vector, and selects the one closest to $\boldsymbol{t}$. It follows that the algorithm selects the lattice vector $\boldsymbol{z}$ such that $\|\boldsymbol{t} - \boldsymbol{z}\|$ is minimized. Since $\|\boldsymbol{t} - \boldsymbol{z}\| < r$ and $r \leq \lambda_1(\mathcal{L})/2$, the closest lattice vector $\boldsymbol{z}$ is unique. Thus, the algorithm correctly finds the lattice vector closest to $\boldsymbol{t}$.

We can now conclude that Algorithm 2 computes the unique lattice vector $\boldsymbol{z} \in \mathcal{L}$ within distance r from the target vector $\boldsymbol{t}$. □

The generalized version of the decoding algorithm for Barnes-Wall lattices correctly finds the closest lattice vector for any lattice of dimension $n = 2^\ell$ with $\ell \in \mathbb{Z}_{\geq 1}$, as long as the hypotheses are satisfied. The decoding algorithm by Micciancio and Nicolosi [MN08] is a recursive application of this general algorithm. The subroutine in the general algorithm is a call to the function itself with as argument the projected target vector.

4.2.3 Complexity

Theorem 24. *For any n-dimensional lattice $\mathcal{L}$ with $n = 2^\ell$ and $\ell \in \mathbb{Z}_{\geq 1}$, Algorithm 2 terminates after $sT_s(n/2) + O(n)$ steps, where s is the number of projections used in the algorithm and $T_s(n/2)$ is the complexity of the subroutine for a problem of size $n/2$.*

Proof. The size of the problem is n and the algorithm solves it by solving s subproblems of size $n/2$. We know that the time complexity is given by:

$$T(n) = sT_s(n/2) + O(s^2 \cdot n),$$

where $T_s(n/2)$ is the time of the subroutine for each subproblem of size $n/2$ and $O(s^2 \cdot n)$ is the time to combine the solutions of the subproblems into a solution of the original problem. We have to reconstruct the original lattice vector for all s^2 pairs and the reconstruction itself is linear. So $O(s^2 \cdot n)$ represents the time to combine the solutions of the subproblems involving $\binom{s}{2}$ pairwise reconstructions, each taking linear time relative to n. Since the number of projections is constant, it follows $O(s^2 \cdot n) = O(n)$. The complexity is then given by:

$$T(n) = sT_s(n/2) + O(n).$$

We cannot apply the masters theorem, because it is not necessarily a recursive application. We cannot further simplify the complexity. □

Remark 25. *In the case of the Barnes-Wall lattices, the algorithm by Micciancio and Nicolosi, is a recursive application of Algorithm 2. The complexity is given by:*

$$T(n) = sT(n/2) + O(n).$$

To determine the time complexity, we apply the masters theorem for divide and conquer recurrences, as given in Theorem 12. To apply the theorem we use $a = s$, $b = 2$ and $d = 1$. It then follows that $\log_b a = log_2(s)$. The Barnes-Wall decoding algorithm uses 4 projections, so $a = 4$. We can conclude that:

$$T(n) = 4T(n/2) + O(n) = O\left(n^{\log_2 4}\right) = O\left(n^2\right),$$

which coincides with the time complexity established for the Barnes-Wall lattice decoding algorithm by Micciancio and Nicolosi [MN08].

4.3 Solving the bounded distance decoding problem in tensor hexagonal lattices

In this section we will apply the general algorithm from 4.2 to the tensor hexagonal lattice from 3.2. Recall that $\psi = \frac{1}{2}\sqrt{3}$.

4.3.1 Hypotheses validation

Before proving that the tensor hexagonal lattice can satisfy the hypotheses for the general algorithm 2, we first introduce some necessary lemmas.

Lemma 26. *Let $\pi_1, \pi_2, \pi_3 : \mathbb{R}^{2n} \to \mathbb{R}^n$ be three projections defined by*

$$\pi_1(\boldsymbol{x}_0, \boldsymbol{x}_1) = \boldsymbol{x}_0, \quad \pi_2(\boldsymbol{x}_0, \boldsymbol{x}_1) = -\tfrac{1}{2}\,\boldsymbol{x}_0 - \psi\,\boldsymbol{x}_1, \quad \pi_3(\boldsymbol{x}_0, \boldsymbol{x}_1) = -\tfrac{1}{2}\,\boldsymbol{x}_0 + \psi\,\boldsymbol{x}_1,$$

where $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{R}^n$. Then for any $i \neq j$, the following holds:

$$\ker(\pi_i) \cap \ker(\pi_j) = \{\mathbf{0}\}.$$

Proof. We are going to proof the lemma by induction on dimension.

Base case:
Let $n = 1$. Then $(x_0, x_1) \in \mathbb{R}^2$ and

$$\pi_1(x_0, x_1) = x_0, \quad \pi_2(x_0, x_1) = -\tfrac{1}{2}\,x_0 - \psi\,x_1, \quad \pi_3(x_0, x_1) = -\tfrac{1}{2}\,x_0 + \psi\,x_1.$$

The kernel equations are given by:

$$\ker(\pi_1) = \{(0, x_1) \mid x_1 \in \mathbb{R}\},$$

$$\ker(\pi_2) = \{(x_0, x_1) \mid -\tfrac{1}{2}x_0 - \psi x_1 = 0\} = \{(2\psi x_1, x_1) \mid x_1 \in \mathbb{R}\},$$

$$\ker(\pi_3) = \{(x_0, x_1) \mid -\tfrac{1}{2}x_0 + \psi x_1 = 0\} = \{(-2\psi x_1, x_1) \mid x_1 \in \mathbb{R}\}.$$

So each $\ker(\pi_i)$ corresponds to a line in $\mathbb{R}^2$. To determine the intersection of two distinct projections, we have to solve the following system:

$$x_0 = 0, \quad -\tfrac{1}{2}x_0 \pm \psi x_1 = 0.$$

It is easy to see that the only solution is $(0, 0)$. Thus, the base case holds.

Inductive step:
Suppose the lemma holds in dimension n. Then for any $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{R}^n$, the intersection of any two kernels is trivial. We will now prove the statement for $2n$. Consider the projections $\pi_1, \pi_2, \pi_3 : \mathbb{R}^{2(2n)} \to \mathbb{R}^{2n}$, defined by:

$$\pi_1(\boldsymbol{x}_0, \boldsymbol{x}_1) = \boldsymbol{x}_0, \quad \pi_2(\boldsymbol{x}_0, \boldsymbol{x}_1) = -\tfrac{1}{2}\,\boldsymbol{x}_0 - \psi\,\boldsymbol{x}_1, \quad \pi_3(\boldsymbol{x}_0, \boldsymbol{x}_1) = -\tfrac{1}{2}\,\boldsymbol{x}_0 + \psi\,\boldsymbol{x}_1,$$

where $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{R}^{2n}$. We need to show that for any $i \neq j$, we have:

$$\ker(\pi_i) \cap \ker(\pi_j) = \{\mathbf{0}\}.$$

Write any $\boldsymbol{z} \in \mathbb{R}^{2(2n)}$ as $(\boldsymbol{x}_0, \boldsymbol{x}_1)$ with $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{R}^{2n}$. The projections π_2 and π_3 introduce linear constraints that apply to each coordinate block of $\boldsymbol{z}$ in the same way as they do in the base case. Specifically, they impose the following system:

$$-\tfrac{1}{2}\boldsymbol{x}_0 - \psi\boldsymbol{x}_1 = \mathbf{0}, \quad -\tfrac{1}{2}\boldsymbol{x}_0 + \psi\boldsymbol{x}_1 = \mathbf{0}.$$

This consists of two independent linear constraints on $(\boldsymbol{x}_0, \boldsymbol{x}_1)$, just as in the $n = 1$ case. Since the same transformation

$$\left(-\tfrac{1}{2}, \pm\psi\right)$$

applies component wise in each $2n$-dimensional sub block and we know with the inductive hypothesis that in each $2n$-dimensional block, the only solution is $\mathbf{0}$, so we can conclude that $\boldsymbol{z} = \mathbf{0}$ in $\mathbb{R}^{2(2n)}$. Thus, $\ker(\pi_i) \cap \ker(\pi_j) = \{\mathbf{0}\}$ in all dimensions by induction. □

Lemma 27. *For the same three projections $\pi_1, \pi_2, \pi_3 : \mathbb{R}^{2n} \to \mathbb{R}^n$, the following holds:*

$$\|\pi_1(\boldsymbol{e})\|^2 + \|\pi_2(\boldsymbol{e})\|^2 + \|\pi_3(\boldsymbol{e})\|^2 = \tfrac{3}{2}\,\|\boldsymbol{e}\|^2$$

for every $\boldsymbol{e} \in \mathbb{R}^{2n}$.

Proof. We are going to proof this by induction on dimension.

Base case:
Let $n = 1$. Then $\boldsymbol{e} = (e_0, e_1) \in \mathbb{R}^2$, and

$$\pi_1(e_0, e_1) = e_0, \quad \pi_2(e_0, e_1) = -\tfrac{1}{2}e_0 - \psi e_1, \quad \pi_3(e_0, e_1) = -\tfrac{1}{2}e_0 + \psi e_1.$$

This gives the following:

$$\begin{aligned}
\|\pi_1(\boldsymbol{e})\|^2 + \|\pi_2(\boldsymbol{e})\|^2 + \|\pi_3(\boldsymbol{e})\|^2 &= \|e_0\|^2 + \| - \tfrac{1}{2}e_0 - \psi e_1\|^2 + \| - \tfrac{1}{2}e_0 + \psi e_1\|^2 \\
&= e_0^2 + \tfrac{1}{4}e_0^2 + \psi^2 e_1^2 + \tfrac{1}{4}e_0^2 + \psi^2 e_1^2 \\
&= e_0^2 + \tfrac{1}{4}e_0^2 + \tfrac{3}{4}e_1^2 + \tfrac{1}{4}e_0^2 + \tfrac{3}{4}e_1^2 \\
&= \tfrac{3}{2}e_0^2 + \tfrac{3}{2}e_1^2 \\
&= \tfrac{3}{2}\,\|\boldsymbol{e}\|^2.
\end{aligned}$$

Inductive step:
Suppose the statement is true in dimension n. We will now prove that it is also true in $2n$. Write $\boldsymbol{e} = (\boldsymbol{e}_0, \boldsymbol{e}_1) \in \mathbb{R}^{2n}$ with each part in $\mathbb{R}^n$. By the same definition as above, we have the following:

$$\pi_1(\boldsymbol{e}_0, \boldsymbol{e}_1) = \boldsymbol{e}_0, \quad \pi_2(\boldsymbol{e}_0, \boldsymbol{e}_1) = -\tfrac{1}{2}\boldsymbol{e}_0 - \psi\boldsymbol{e}_1, \quad \pi_3(\boldsymbol{e}_0\boldsymbol{e}_1) = -\tfrac{1}{2}\boldsymbol{e}_0 + \psi\boldsymbol{e}_1.$$

It then follows that:

$$\begin{aligned}\|\pi_1(\boldsymbol{e})\|^2 + \|\pi_2(\boldsymbol{e})\|^2 + \|\pi_3(\boldsymbol{e})\|^2 &= \|\boldsymbol{e}_0\|^2 + \| - \tfrac{1}{2}\boldsymbol{e}_0 - \psi\boldsymbol{e}_1\|^2 + \| - \tfrac{1}{2}\boldsymbol{e}_0 + \psi\boldsymbol{e}_1\|^2 \\ &= \tfrac{3}{2}\left(\|\boldsymbol{e}_0\|^2 + \|\boldsymbol{e}_1\|^2\right) \\ &= \tfrac{3}{2}\|\boldsymbol{e}\|^2.\end{aligned}$$

Hence the statement is true in dimension $2n$, completing the induction. □

We can now proof that the hypotheses of the general algorithm apply to the tensor hexagonal lattice. We introduce the following theorem.

Theorem 28. *For any integer $\ell \geq 1$, the $n = 2^\ell$ dimensional tensor hexagonal lattice TH_ℓ satisfies the four hypotheses required for the general decoding algorithm 2 with 3 projections defined by:*

$$\pi_1(\boldsymbol{t}) = \boldsymbol{t}_0, \quad \pi_2(\boldsymbol{t}) = -\tfrac{1}{2}\boldsymbol{t}_0 - \psi\boldsymbol{t}_1, \quad \pi_3(\boldsymbol{t}) = -\tfrac{1}{2}\boldsymbol{t}_0 + \psi\boldsymbol{t}_1,$$

with $\boldsymbol{t} \in \mathbb{R}^n$ and $\gamma^2 = 3/4$. Consequently, the general decoding algorithm can be successfully applied to TH_ℓ to solve the BDD problem within the unique decoding radius.

Proof. We will proof this theorem with induction on ℓ.

For $\ell = 1$ we have the hexagonal lattice of dimension 2. We will now verify the four hypothesis for TH_1.

(H1) Let $\boldsymbol{z} \in TH_1$. Then:

$$\boldsymbol{z} = x_1\left(1, 0\right) + x_2\left(\tfrac{1}{2}, \psi\right) = \left(x_1 + \tfrac{1}{2}x_2, \psi x_2\right), \quad x_1, x_2 \in \mathbb{Z}.$$

By substituting this into the projections π_1, π_2 and π_3 we obtain:

$$\pi_1(\boldsymbol{z}) = x_1 + \tfrac{1}{2}x_2, \quad \pi_2(\boldsymbol{z}) = -\tfrac{1}{2}\left(x_1 + \tfrac{1}{2}x_2\right) - \psi(\psi x_2) = -\tfrac{1}{2}x_1 - x_2,$$

$$\pi_3(\boldsymbol{z}) = -\tfrac{1}{2}\left(x_1 + \tfrac{1}{2}x_2\right) + \psi(\psi x_2) = -\tfrac{1}{2}x_1 + \tfrac{1}{2}x_2.$$

Since $x_1, x_2 \in \mathbb{Z}$, we can conclude that:

$$\pi_1(TH_1) = \mathbb{Z} + \frac{1}{2}\mathbb{Z} \simeq \mathbb{Z}, \quad \pi_2(TH_1) = -\tfrac{1}{2}\mathbb{Z} - \mathbb{Z} \simeq \mathbb{Z}, \quad \pi_3(TH_1) = -\tfrac{1}{2}\mathbb{Z} + \tfrac{1}{2}\mathbb{Z} \simeq \mathbb{Z}.$$

Thus, each projection maps TH_1 isomorphically to $\mathcal{L}' = \mathbb{Z}$, satisfying Hypothesis 1.

(H2) By Lemma 26 in the base case $n = 1$, we have $\ker(\pi_i) \cap \ker(\pi_j) = \{\boldsymbol{0}\}$ for $i \neq j$, so Hypothesis 2 is fulfilled.

(H3) Using Lemma 27, we have:

$$\|\pi_1(\boldsymbol{e})\|^2 + \|\pi_2(\boldsymbol{e})\|^2 + \|\pi_3(\boldsymbol{e})\|^2 = \tfrac{3}{2}\|\boldsymbol{e}\|^2.$$

If fewer than two of these are $\leq \gamma \cdot \|\boldsymbol{e}\|$ for $\gamma = \frac{\sqrt{3}}{2}$, the sum of squares would exceed $\frac{3}{2}\|\boldsymbol{e}\|^2$. This is a contradiction with Lemma 27. We can conclude that at least two lie below $\gamma \cdot \|\boldsymbol{e}\|$, fulfilling (H3).

(H4) Each of the projections maps TH_1 to $\mathcal{L}' = \mathbb{Z}$. The bounded distance decoding problem in $\mathbb{Z}$ can be solved by rounding the target vector to the closest integer, as long as the distance between the target vector and the lattice vector is less than $\frac{1}{2}$ (otherwise we aren't guaranteed unique solutions). So we have a decoding algorithm for a decoding radius $r = \frac{1}{2}$, which satisfies the condition

$$\gamma \cdot r = \tfrac{\sqrt{3}}{2} \cdot \tfrac{1}{2} = \tfrac{\sqrt{3}}{4} < \tfrac{1}{2} = \tfrac{\lambda_1(\mathbb{Z})}{2}.$$

Thus, there exists a decoding algorithm for $\mathcal{L} = \mathbb{Z}$ that operates within the required radius, satisfying Hypothesis 4.

We can now conclude that the theorem holds for the tensor hexagonal lattice of dimension 2.

Next we will proof that if there exists a $k \geq 1$ for which the hypotheses are satisfied, the hypotheses are also satisfied in the tensor hexagonal lattice of dimension 2^{k+1}. We will again proof them one by one.

(H1) Any lattice point $\boldsymbol{z} \in TH_{k+1}$ can be expressed as:

$$\boldsymbol{z} = \left(\boldsymbol{u} + \tfrac{1}{2}\boldsymbol{v}, \psi\boldsymbol{v}\right),$$

where $\boldsymbol{u}, \boldsymbol{v} \in TH_k$. Applying the projections yields the following results:

$$\pi_1(\boldsymbol{z}) = \boldsymbol{u} + \tfrac{1}{2}\boldsymbol{v} \in TH_k + \tfrac{1}{2}TH_k \simeq TH_k,$$

$$\pi_2(\boldsymbol{z}) = -\tfrac{1}{2}\boldsymbol{z}_0 - \psi\boldsymbol{z}_1 = -\tfrac{1}{2}(\boldsymbol{u} - \tfrac{1}{2}\boldsymbol{v}) + \psi(\psi\boldsymbol{v}) = -\tfrac{1}{2}\boldsymbol{u} - \tfrac{1}{4}\boldsymbol{v} - \tfrac{3}{4}\boldsymbol{v} = -\tfrac{1}{2}\boldsymbol{u} - \boldsymbol{v} \in \tfrac{1}{2}TH_k + TH_k \simeq TH_k,$$

$$\pi_3(\boldsymbol{z}) = -\tfrac{1}{2}\boldsymbol{z}_0 + \psi\boldsymbol{z}_1 = -\tfrac{1}{2}(\boldsymbol{u} + \tfrac{1}{2}\boldsymbol{v}) + \psi(\psi\boldsymbol{v}) = -\tfrac{1}{2}\boldsymbol{u} + \tfrac{1}{2}\boldsymbol{v} \simeq TH_k.$$

So each projection $\pi_i(\boldsymbol{z})$ maps TH_{k+1} isomorphically to TH_k, thereby satisfying Hypothesis 1 (the lower dimensional tensor hexagonal lattice has half the dimension of the original tensor hexagonal lattice).

(H2) By Lemma 26 in dimension $2n$, the kernels of any two distinct projections π_i, π_j intersect only in $\{\mathbf{0}\}$. Thus Hypothesis 2 is satisfied.

(H3) By Lemma 27 in dimension $2n$, for any $\boldsymbol{e} \in \mathbb{R}^{2n}$, the following holds:

$$\|\pi_1(\boldsymbol{e})\|^2 + \|\pi_2(\boldsymbol{e})\|^2 + \|\pi_3(\boldsymbol{e})\|^2 = \tfrac{3}{2}\|\boldsymbol{e}\|^2,$$

forcing at least two projections below $\gamma\|\boldsymbol{e}\|$ with $\gamma = \frac{\sqrt{3}}{2}$. Therefore Hypothesis 3 holds.

(H4) Each projection maps TH_{k+1} to TH_k, which by the inductive hypothesis can be decoded within radius $\gamma\, r \leq \lambda_1(TH_k)/2$. Thus Hypothesis 4 is satisfied.

Having verified all four hypotheses for TH_{k+1}, the inductive step is complete. We can conclude that the theorem holds for all tensor hexagonal lattices TH_k of dimension 2^k with $k \geq 1$. □

4.3.2 Decoding algorithm

We can now give the algorithm for solving the bounded distance decoding problem in the tensor hexagonal lattice of dimension k. Using the hypotheses we can explicitly reconstruct the original lattice vector. With Hypothesis 2 we know that any error vector can be recovered correctly form any two projections. Let $\left(\boldsymbol{z}_-, \boldsymbol{z}_+\right) = -\left(1/2\right) \cdot \left(\boldsymbol{z}_0 + \boldsymbol{z}_1\sqrt{3}, \boldsymbol{z}_0 - \boldsymbol{z}_1\sqrt{3}\right)$. At least two projections can be decoded correcly, we need to show that we can recover the original lattice vector from any of the three pairs $\left(\boldsymbol{z}_0, \boldsymbol{z}_+\right)$, $\left(\boldsymbol{z}_0, \boldsymbol{z}_-\right)$ or $\left(\boldsymbol{z}_-, \boldsymbol{z}_+\right)$.

Assume that we know $\boldsymbol{z}_0$ and we want to recover $\boldsymbol{z}_1$ when knowing $\boldsymbol{z}_+$. We get the following:

$$\boldsymbol{z}_+ = -\tfrac{1}{2}\boldsymbol{z}_0 + \tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} \;\Leftrightarrow\; \tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} = \boldsymbol{z}_+ + \tfrac{1}{2}\boldsymbol{z}_0 \;\Leftrightarrow\; \boldsymbol{z}_1 = \tfrac{2}{\sqrt{3}}\boldsymbol{z}_+ + \tfrac{1}{\sqrt{3}}\boldsymbol{z}_0.$$

Assume that we know $\boldsymbol{z}_0$ and we want to recover $\boldsymbol{z}_1$ when knowing $\boldsymbol{z}_-$. We get the following:

$$\boldsymbol{z}_- = -\tfrac{1}{2}\boldsymbol{z}_0 - \tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} \;\Leftrightarrow\; -\tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} = \boldsymbol{z}_- + \tfrac{1}{2}\boldsymbol{z}_0 \;\Leftrightarrow\; \boldsymbol{z}_1 = -\tfrac{2}{\sqrt{3}}\boldsymbol{z}_- - \tfrac{1}{\sqrt{3}}\boldsymbol{z}_0.$$

Assume that we know $\boldsymbol{z}_+$ and $\boldsymbol{z}_-$ and we want to recover $\boldsymbol{z}_0$ and $\boldsymbol{z}_1$. We know the following:

$$\boldsymbol{z}_+ = -\tfrac{1}{2}\boldsymbol{z}_0 + \tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} \;\Leftrightarrow\; \tfrac{1}{2}\boldsymbol{z}_0 = \tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} - \boldsymbol{z}_+ \;\Leftrightarrow\; \boldsymbol{z}_0 = \boldsymbol{z}_1\sqrt{3} - 2\boldsymbol{z}_+$$

and

$$\boldsymbol{z}_- = -\tfrac{1}{2}\boldsymbol{z}_0 - \tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} \;\Leftrightarrow\; \tfrac{1}{2}\boldsymbol{z}_0 = -\boldsymbol{z}_- - \tfrac{1}{2}\boldsymbol{z}_1\sqrt{3} \;\Leftrightarrow\; \boldsymbol{z}_0 = -\boldsymbol{z}_1\sqrt{3} - 2\boldsymbol{z}_-.$$

It then follows that:

$$\boldsymbol{z}_1\sqrt{3} - 2\boldsymbol{z}_+ = \boldsymbol{z}_0 = -\boldsymbol{z}_1\sqrt{3} - 2\boldsymbol{z}_- \;\Leftrightarrow\; 2\boldsymbol{z}_1\sqrt{3} = 2\boldsymbol{z}_+ - 2\boldsymbol{z}_- \;\Leftrightarrow\; \boldsymbol{z}_1 = \tfrac{1}{\sqrt{3}}\boldsymbol{z}_+ - \tfrac{1}{\sqrt{3}}\boldsymbol{z}_-.$$

Substituting this into $\boldsymbol{z}_0 = \boldsymbol{z}_1\sqrt{3} - 2\boldsymbol{z}_+$ gives us the following:

$$\boldsymbol{z}_0 = \sqrt{3}\cdot\left(\tfrac{1}{\sqrt{3}}\boldsymbol{z}_+ - \tfrac{1}{\sqrt{3}}\boldsymbol{z}_-\right) - 2\boldsymbol{z}_+ - = \boldsymbol{z}_+ - \boldsymbol{z}_- - 2\boldsymbol{z}_+ = -\boldsymbol{z}_+ - \boldsymbol{z}_-.$$

The explicit reconstruction of the original lattice vector can be found in the pseudocode for the algorithm, which is given below.

Algorithm 3 Decoding algorithm for tensor hexagonal lattices of dimension $n = 2^k$

function DECODETH($\boldsymbol{t}$)
  if $\boldsymbol{t} \in \mathbb{R}^1$ **then**
    return $\lceil \boldsymbol{t} \rfloor$ — Round $\boldsymbol{t}$ to the closest integer
  else
    $\left(\boldsymbol{t}_0, \boldsymbol{t}_1\right) \leftarrow \boldsymbol{t}$ — Split $\boldsymbol{t}$ into two halves
    $\left(\boldsymbol{t}_-, \boldsymbol{t}_+\right) \leftarrow -(1/2)\cdot\left(\boldsymbol{t}_0 + \boldsymbol{t}_1\sqrt{3}, \boldsymbol{t}_0 - \boldsymbol{t}_1\sqrt{3}\right)$
    for all $i \in \{0, +, -\}$ **do**
      $\boldsymbol{z}_i \leftarrow$ DECODETH($\boldsymbol{t}_i$) — Decode the projections in the projected lattice
    end for
    $\boldsymbol{z}_0^+ \leftarrow \left(\boldsymbol{z}_0, \tfrac{2}{\sqrt{3}}\boldsymbol{z}_+ + \tfrac{1}{\sqrt{3}}\boldsymbol{z}_0\right)$ — Compute the 3 candidate vectors $\left(\boldsymbol{z}_0, \boldsymbol{z}_1\right) \in TH_k$
    $\boldsymbol{z}_0^- \leftarrow \left(\boldsymbol{z}_0, -\tfrac{2}{\sqrt{3}}\boldsymbol{z}_- - \tfrac{1}{\sqrt{3}}\boldsymbol{z}_0\right)$
    $\boldsymbol{z}_-^+ \leftarrow \left(-\boldsymbol{z}_+ - \boldsymbol{z}_-, \tfrac{1}{\sqrt{3}}\boldsymbol{z}_+ - \tfrac{1}{\sqrt{3}}\boldsymbol{z}_-\right)$
    $\boldsymbol{z} \leftarrow \arg\min_{\boldsymbol{z}' \in \{\boldsymbol{z}_0^+, \boldsymbol{z}_0^-, \boldsymbol{z}_-^+\}}\{\|\boldsymbol{t} - \boldsymbol{z}'\|\}$ — Select the candidate closest to $\boldsymbol{t}$
    return $\boldsymbol{z}$
  end if
end function

4.3.3 Correctness and complexity

The correctness of the algorithm follows directly from the fact that the hypotheses are satisfied.

We use three projections, so using the formula for complexity for the general algorithm, we can conclude that the the complexity is:

$$O(n^{\log_2(3)}) \equiv O(n^{1.5850}).$$

Chapter 5

Conclusion and further research

The goal of this thesis was to explore the bounded distance decoding (BDD) problem and efficient solutions for the problem within specific lattice structures. We investigated efficient solutions for the Barnes-Wall lattices and generalized this approach to an efficient solution for tensor hexagonal lattices. We were able to determine the time complexity for solving the BDD problem in both cases and we determined the normalized minimum distance for both lattice families. Recall that we are ideally looking for lattices where we can solve the BDD problem efficiently with some trapdoor information and for lattices with large minimum distance.

For Barnes-Wall lattices, we demonstrated that the BDD problem can be solved in $\mathcal{O}\left(n^2\right)$ time. For tensor hexagonal lattices, we demonstrated that the BDD problem can be solved in $\mathcal{O}\left(n^{1.5850}\right)$ time.

We determined that the normalized minimum distance for Barnes-Wall lattices is $\Theta\left(n^{1/4}\right)$. We also determined that the normalized minimum distance for tensor hexagonal lattices is $\Theta\left(n^{0.1038}\right)$.

We conclude that although the BDD problem is solved faster in tensor hexagonal lattices, the normalized minimum distance is larger in Barnes-Wall lattices compared to tensor hexagonal lattices.

Our findings raise several questions for future research. Is it possible to improve efficiency of the tensor hexagonal lattice decoding process? Can the efficiency of the general BDD algorithm proposed in this thesis be improved? And, can this general algorithm be applied to other lattice families to further extend its applicability?

Bibliography

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms (3rd edition). *MIT Press and McGraw-Hill*, 2009.

[Cor20] Vincent Corlay. *Decoding Algorithms for Lattices*. PhD thesis, Institut Polytechnique de Paris, 2020.

[DD18a] D. Dadush and L. Ducas. Introduction to Lattice Algorithms and Cryptography, Lecture 1 – Introduction. https://homepages.cwi.nl/~dadush/teaching/lattices-2018/notes/lecture-1.pdf, 2018.

[DD18b] D. Dadush and L. Ducas. Introduction to Lattice Algorithms and Cryptography, Lecture 2 – Determinant, Packing and Covering, and the Minkowski Theorems. https://homepages.cwi.nl/~dadush/teaching/lattices-2018/notes/lecture-2.pdf, 2018.

[DP19] Léo Ducas and Cécile Pierrot. Polynomial Time Bounded Distance Decoding near Minkowski's Bound in Discrete Logarithm Lattices. *Designs, Codes and Cryptography*, 2019.

[FA24] Efat Fathalla and Mohamed Azab. Beyond Classical Cryptography: A Systematic Review of Post-Quantum Hash-Based Signature Schemes, Security, and Optimizations. *IEEE Access*, 2024.

[For88] G David Forney. Coset codes - Part I: Introduction and Geometrical Classification. *IEEE Transactions on Information Theory*, 34(5):1123–1151, 1988.

[GP12] Elena Grigorescu and Chris Peikert. List decoding barnes-wall lattices. In *2012 IEEE 27th Conference on Computational Complexity*, pages 316–325. IEEE, 2012.

[HWL08] Jr. Hendrik W. Lenstra. Lattices. https://www.math.leidenuniv.nl/~stevenhagenp//ANTproc/06hwl.pdf, 2008.

[Lap21] Oleksandra Lapiha. Comparing Lattice Families for Bounded Distance Decoding near Minkowskis Bound. *Cryptology ePrint Archive*, 2021.

[LSLY] Zhe Li, Chaoping Xing San Ling, and Sze Ling Yeo. On the Bounded Distance Decoding Problem for Lattices Constructed from Polynomials and Their Cryptographic Applications.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a Cryptographic Perspective*, volume 671. Springer Science & Business Media, 2002.

[MN08] Daniele Micciancio and Antonio Nicolosi. Efficient bounded distance decoders for Barnes-Wall lattices. In *2008 IEEE International Symposium on Information Theory*. IEEE, 2008.

[vP16] Alex van Poppelen. Cryptographic decoding of the Leech lattice. Master's thesis, 2016.

[vW23] Wessel Pieter Jacobus van Woerden. *Lattice Cryptography, from Cryptanalysis to New Foundations*. PhD thesis, Leiden University, 2023.

Appendix A

A.1 Equivalence of lattices over $\mathbb{G}$ and lattices over $\mathbb{Z}$

There exists an isomorphism between any n-dimensional real lattice and a corresponding $2n$-dimensional lattice over the integers $\mathbb{Z}$ [For88]. This isomorphism turns each complex number into a pair of coordinates in the real lattice. The isomorphism is given by:

$$f : a + bi \mapsto \big(a, b\big) \quad a, b \in \mathbb{Z}.$$

Let $\mathbf{B} \in \mathbb{G}^{n\times n}$ be a lattice basis with rows in $\mathbb{G}^n$. Then, there exists a basis matrix $\mathbf{B'} \in \mathbb{Z}^{2n\times 2n}$ such that

$$\{\boldsymbol{x} \cdot \mathbf{B} : \boldsymbol{x} \in \mathbb{G}^n\} = \mathcal{L}(\mathbf{B}) \cong \mathcal{L}(\mathbf{B'}) = \left\{\boldsymbol{y} \cdot \mathbf{B'} : \boldsymbol{y} \in \mathbb{Z}^{2n}\right\},$$

where the isomorphism preserves lattice structure. However, it does not preserve the multiplicative structure of the ambient spaces [For88]. Therefore, while the ambient spaces $\mathbb{G}^n$ and $\mathbb{Z}^{2n}$ are not isomorphic, the lattices defined over them are isomorphic. We can conclude that n-dimensional lattices described by bases over the Gaussian integers are equivalent to $2n$-dimensional lattices described by bases over the integers.

The simplest example of a complex lattice is the one-dimensional complex lattice $\mathbb{G}$ corresponding to the two-dimensional real lattice $\mathbb{Z}^2$ [For88]. The point $\big(a, b\big)$ in $\mathbb{Z}^2$ corresponds to the point $a + bi$ in $\mathbb{G}$, where a and b may be any pair of integers [For88].

A.2 Implementation decoder

```
"""
Implementation of a modified version of the parallel bounded distance decoder
(BDD) for Barnes-Wall lattices, given in Algorithm 1 in the paper "Efficient
Bounded Distance Decoders for Barnes-Wall Lattices" by Daniele Micciancio and
Antonio Nicolosi. It also includes a test function to see how the algorithm
performs inside and outside the squared unique decoding radius.
"""

import matplotlib.pyplot as plt
import numpy as np
from multiprocessing import Pool
from time import time
from numpy.random import seed

phi = 1 + 1j # Prime of the least squared norm in the Gaussian integers
```

```
"""
Bounded Distance Decoder (BDD) for Barnes-Wall lattices
For any N = 2^n and s in C^N such that dist^2(s, BW^n) < N/4, Algorithm 1 computes
the (unique) lattice vector z in BW^n within squared distance N/4 from the target
vector s.
"""

def bounded_distance_decoder(s):

    # If s is a one dimensional complex vector, we round s component wise to the
        closest Gaussian integer.
    if len(s) == 1:
        return np.round(s.real) + 1j * np.round(s.imag)

    else:
        s_0, s_1 = np.split(s, 2)  # Split the target vector into two halves

        # Apply the automorphism to the target vector
        s_minus = (phi / 2) * (s_0 - s_1)
        s_plus = (phi / 2) * (s_0 + s_1)

        # Execute recursive calls
        z_0 = bounded_distance_decoder(s_0)
        z_1 = bounded_distance_decoder(s_1)
        z_minus = bounded_distance_decoder(s_minus)
        z_plus = bounded_distance_decoder(s_plus)

        # Compute the 4 candidate vectors
        z_0_minus = np.concatenate([z_0, z_0 - 2 * (1 / phi) * z_minus])
        z_0_plus = np.concatenate([z_0, 2 * (1 / phi) * z_plus - z_0])
        z_1_minus = np.concatenate([2 * (1 / phi) * z_minus + z_1, z_1])
        z_1_plus = np.concatenate([2 * (1 / phi) * z_plus - z_1, z_1])

        candidates = [z_0_minus, z_0_plus, z_1_minus, z_1_plus]
        # Select the candidate closest to s
        z = min(candidates, key=lambda candidate: np.linalg.norm(s - candidate))
        return z

"""
Generate the basis matrix for the Barnes-Wall lattice of dimension 2^n.
"""

def generate_lattice(n):

    # Matrix used to generate higher dimensional generator matrices
    base_matrix_BW = np.array([[1, 1], [0, phi]])

    if n == 0:
        return np.array([1])
    elif n == 1:
        return base_matrix_BW
    else:
        BW_power_n = base_matrix_BW
        for _ in range(n - 1):
            # Calculate the n fold Kronicker product of the base matrix
            BW_power_n = np.kron(BW_power_n, base_matrix_BW)
        return BW_power_n


"""
Function that generates a random vector in the Barnes-Wall lattice. A lattice
vector is a gaussian linear integer combination of the rows of the basis matrix.
"""
```

```
def random_BW_vector(n):

    # Basis matrix for 2^n dimensional Barnes-Wall lattice
    BW_power_n_matrix = generate_lattice(n)
    coefficients_real = np.random.randint(-100, 100, size=2 ** n)
    coefficients_im = np.random.randint(-100, 100, size=2 ** n)
    # Random vector that is a combination of the rows of A
    random_vector = np.dot(coefficients_real + 1j * coefficients_im,
        BW_power_n_matrix)
    return random_vector

"""
Given a lattice point in a N = 2^n dimensional Barnes-Wall lattice, generate a
vector s at a specified distance from this lattice point. For alpha = 1 we get
points exactly on the unique decoding radius, for alpha > 1 we get points
outside the squared unique decoding radius and for alpha < 1 we get inside the
squared unique decoding radius.
"""

def generate_s(n, z, alpha):

    N = 2 ** n
    unique_radius_squared = N/4
    squared_distance = alpha * unique_radius_squared
    # Create the perturbation vector with each coordinate a Gaussian random variable.
    # Normal distribution ensures the perturbations are spread in all directions.
    delta = np.random.normal(0, 1, N) + 1j * np.random.normal(0, 1, N)
    delta_normalized = delta / np.linalg.norm(delta)
    delta_scaled = delta_normalized * np.sqrt(squared_distance)
    s = z + delta_scaled
    return s

"""
Test the algorithm's performance at various distances alpha * N/4 from the
lattice. Input is a list of alpha values, and a number of trials for each
alpha value.
"""

def aux(params):

    n, alpha, trials = params
    seed(int(np.round(alpha*1e8)))
    successes = 0
    for _ in range(trials):
        z = random_BW_vector(n)
        s = generate_s(n, z, alpha)
        decoded_z = bounded_distance_decoder(s)
        # Check if the decoded result matches the original lattice point
        if np.array_equal(decoded_z, z):
            successes += 1
    success_rate = successes / trials
    return (alpha, success_rate)


def test_algorithm(n, alpha_values, trials, cores=32):
    with Pool(cores) as p:
        results = p.map(aux, [(n, a, trials) for a in alpha_values])
    return results


for n in range(2, 13):
    T0 = time()
    print("starting experiment %d"%n)
    L = test_algorithm(n, np.linspace(0, 4, 101), 1000)
```

```
file = open("exp_n%d.txt"%n, "w")
file.write("n, \t alpha, \t succ_rate \n")
for (alpha, sr) in L:
    file.write("%d, \t %.4f, \t %.4f \n"%(n, alpha, sr))
file.close()
T1 = time()
print("Done in Time %.2f s"%(T1-T0))
```