

Quantum and Classical Algorithms for Bounded Distance Decoding

Richard Allen* Ratip Emin Berker† Sílvia Casacuberta‡ Michael Gul§

Harvard University

February 17, 2022

Abstract

In this paper, we provide a comprehensive overview of a recent debate over the quantum versus classical solvability of bounded distance decoding (BDD). Specifically, we review the work of Eldar and Hallgren [EH22], [Hal21] demonstrating a quantum algorithm solving $\lambda_1 2^{-\Omega(\sqrt{k \log q})}$ -BDD in polynomial time for lattices of periodicity q , finite group rank k , and shortest lattice vector length λ_1 . Subsequently, we prove the results of [DvW21a], [DvW21b] with far greater detail and elaboration than in the original work. Namely, we show that there exists a deterministic, classical algorithm achieving the same result.

Contents

1	Introduction	2
1.1	Timeline of Publications	2
1.2	Outline of this Paper	3
2	Background on Lattices, Relations, and Reductions	3
2.1	Definitions and Notation	3
2.2	SVP, CVP, and BDD	4
2.3	Lattice Hardness Results	5
2.4	Survey of Quantum Approaches to Lattice Problems	6
3	Overview of Eldar and Hallgren’s Algorithm	6
3.1	Quantum Primitives	7
3.2	Intuition for the Proof	7
4	Classical Primitives	8
4.1	The LLL Algorithm	8
4.2	Babai’s Nearest Plane Algorithm	10
5	Proof of Classical Alternative	11
5.1	Short Bases from Short Full-Rank Sets	11
5.2	Duality	12
5.3	Completing the Proof	14
6	Conclusion and Future Work	16

*rallen@college.harvard.edu

†rberker@college.harvard.edu

‡scasacubertapuig@college.harvard.edu

§michaelgul@college.harvard.edu

1 Introduction

Lattice-based encryption, and specifically encryptions based on the Learning With Errors (LWE) problem as introduced by Regev [Reg10], are considered a promising candidate for post-quantum encryption. The security of LWE rests on the conjectured hardness of lattice problems such as the Closest Vector Problem (CVP) and Bounded Distance Decoding (BDD). However, Eldar and Hallgren [Hal21] have recently proposed a quantum algorithm for BDD with a subexponential approximation factor. The specific quantum protocol used to achieve this exponential speed-up raised questions on the purely classical achievability of the same result. Specifically, Eldar and Hallgren’s algorithm uses the Quantum Phase Estimation (QPE) protocol to reduce a worst-case instance of BDD to a random instance in a lattice of lower dimension, while preserving the bound on error magnitude. The resulting problem has a known classical solution. Since the core quantum step is the use of QPE for dimensionality reduction, a natural question is whether classical dimensionality reduction algorithms could be substituted. Shortly after this result, Ducas and van Woerden [DvW21a] showed that the Lenstra-Lenstra-Lovász (LLL) lattice basis reduction algorithm can be used to achieve the same result for a specific class of lattices, later generalizing this result to the same type of lattice considered by Eldar and Hallgren. In this work, we give an overview of Eldar and Hallgren’s quantum algorithm, followed by a detailed exposition of Ducas and van Woerden’s classical algorithm. In order to develop the necessary tools for the proof, we survey the field of lattice problems in general, and in particular describe the LLL algorithm in detail.

1.1 Timeline of Publications

Eldar and Hallgren first presented their algorithm in a talk given by Hallgren on September 21, 2021 at the Simons Institute [Hal21]. Their main point can be summarized as follows:

Theorem 1.1 (Eldar and Hallgren [EH22]). *There exists a $\text{poly}(n, \log q)$ -time quantum algorithm solving $\lambda_1 2^{-\Omega(\sqrt{k \log q})}$ -BDD on lattices of dimension n , periodicity q , and finite group rank k with shortest lattice vector length λ_1 .*

We elaborate on the definitions and implications of the parameters n, q , and k further below. While Hallgren’s talk only considered the special case $q = 2^n, k = 1$, it was also briefly mentioned that the theorem holds for arbitrary q and k . Three days after the talk, on September 24, 2021, Ducas and van Woerden published a note [DvW21a] proving that known classical polynomial-time algorithms (namely, LLL and Babai’s algorithms) are sufficient to achieve the result presented by Eldar and Hallgren. Similar to Hallgren’s talk, the note only considered the special case where $q = 2^n, k = 1$, hence presenting the theorem:

Theorem 1.2 (Ducas and van Woerden, Version 1 [DvW21a]). *For any given vector $a \in \mathbb{Z}^n$, define the lattice $L_a = q\mathbb{Z}^n + a\mathbb{Z}$. There exists a deterministic polynomial-time algorithm that solves BDD in L_a for any error up to radius $\lambda_1(L_a) \cdot 2^{-\Theta(\sqrt{n})}$, where $\lambda_1(L_a)$ is the length of the shortest vector in L_a .*

Given that Hallgren had mentioned in his talk that their result is generalizable to arbitrary q and k , this motivated us to work on generalizing Ducas and van Woerden’s proof to arbitrary q and k as well. Following this, we received Eldar and Hallgren’s preprint through private correspondence, in which their result was indeed proven for arbitrary q and k [EH22]. While we were working independently on a proof of the classical algorithm for arbitrary q and k , we realized later that on October 14, 2021, Ducas and van Woerden had published a second version of their note [DvW21b], explaining that their result is generalizable for arbitrary q and k :

Theorem 1.3 (Ducas and van Woerden, Version 2 [DvW21b]). *For any given matrix $\mathbf{A} \in \mathbb{Z}^{n \times k}$, define the lattice $L_{\mathbf{A}} = q\mathbb{Z}^n + \mathbf{A}\mathbb{Z}^k$. There exists a deterministic polynomial-time algorithm that solves BDD in $L_{\mathbf{A}}$ for any error up to radius $\frac{1}{2}\lambda_1(L_{\mathbf{A}}) \cdot \exp\{-\Omega(\sqrt{k} \cdot \ln q)\}$.*

Note that in principle, \mathbf{A} is arbitrary, but in order for k to give the finite group rank we generally restrict the columns of A to be linearly independent over \mathbb{Z} . From this point on, we sought to elaborate on their result, filling in missing details and unifying a summary of Eldar and Hallgren’s algorithm with an elaboration on Ducas and van Woerden’s proof for the general case.

1.2 Outline of this Paper

This paper is structured as follows. In Section 2, we introduce the necessary definitions and tools to work with lattices (Section 2.1), as well as the main computational lattice problems that we require for the discussion, mainly SVP, CVP, and BDD (Section 2.2), along with their known hardness results and reductions (Section 2.3). We also provide a survey on other important lattice-based quantum algorithms beyond the result of Eldar and Hallgren (Section 2.4). In Section 3, we provide an overview of the quantum algorithm by Eldar and Hallgren, including the necessary quantum primitives (Section 3.1) and the intuition for their proof (Section 3.2). In Section 4, we review the main two classical lattice algorithms, namely the LLL algorithm (Section 4.1) and Babai’s nearest plane algorithm (Section 4.2).

As we describe the algorithms and their correctness, we extract the necessary facts for the proof of the classical counterpart to Eldar and Hallgren’s quantum algorithm, used in Section 5. Our main proof is split into three parts: we first show how to obtain short bases from full-rank sets of short lattice vectors (Section 5.1), we then turn to dual lattices and their relevant properties (Section 5.2), and lastly we put together all the ingredients of the proof (Section 5.3).

2 Background on Lattices, Relations, and Reductions

2.1 Definitions and Notation

Before turning our attention to BDD, we survey a number of lattice problems and the connections between them. The basic objects of interest are integer lattices:

Definition 2.1 (Lattice). A lattice \mathcal{L} is generated by all the integer combinations of the vectors of some basis \mathbf{B} :

$$\mathcal{L} = \sum_{i=1}^m \mathbb{Z}b_i = \left\{ \sum_{i=0}^m z_i b_i, \text{ where } z_i \in \mathbb{Z}, b_i \in \mathbf{B} \right\}.$$

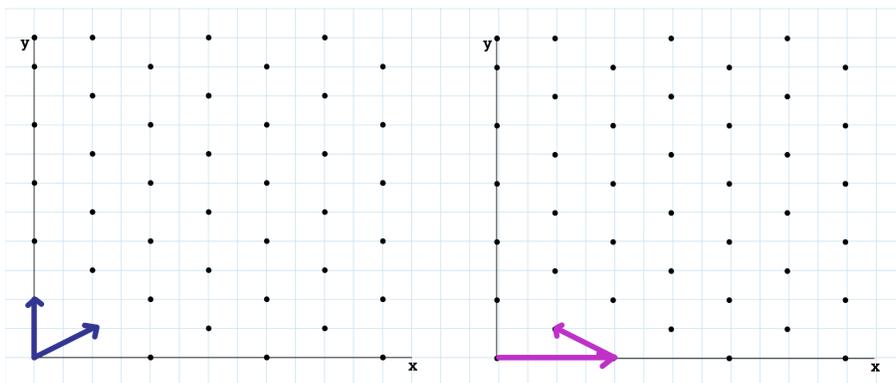


Figure 1: An example of a lattice in \mathbb{R}^2 , with two bases given which correspond to $\{[2, 1], [0, 2]\}$ and $\{[4, 0], [-2, 1]\}$.

Lattices have infinitely many bases, and so it is natural to ask which bases are *better* in some sense. We will see that the two main properties we will consider are that the basis vectors are as *short* and as *orthogonal* as possible. In fact, one of the key insights guiding shortest vector algorithms is that the latter of these properties implies the former. We will discuss this further in Section 4, when we introduce the LLL algorithm.

The volume of the lattice is denoted by $\det(\mathcal{L})$. For any basis $\mathbf{B} = \{b_1, \dots, b_n\}$ for \mathcal{L} , the determinant of \mathcal{L} is defined by

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

where \mathbf{B} is understood as the basis matrix constructed with the vectors b_i as columns. If \mathcal{L} is full-rank, then \mathbf{B} is a square matrix, so we can also think of the volume of the lattice as $|\det(\mathbf{B})|$.

Geometrically, the determinant of the lattice is the volume of the parallelepiped spanned by the b_i 's [Bai06]. It is important to remark that the volume of the lattice is an invariant; that is, does not depend on the choice of the basis. This is because any two different bases $\mathbf{B}_1, \mathbf{B}_2$ for the same lattice \mathcal{L} are related by a unimodular matrix [Bai06]. Specifically, there exists a matrix U with $\det(U) = \pm 1$ such that

$$\mathbf{B}_1 = U \cdot \mathbf{B}_2.$$

Hence it follows that $|\det(\mathbf{B}_1)| = |\det(\mathbf{B}_2)|$.

We will return to the topic of volume of lattices in Section 4, when we further analyze the properties of $\det(\mathcal{L})$.

2.2 SVP, CVP, and BDD

Long before lattices were used in cryptography, computational problems in the field of lattice theory had been considered in other contexts. The two main ones are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP):

Definition 2.2 (γ -SVP [DD18]). Given a lattice $\mathcal{L} \subset \mathbb{R}^n$, find a lattice vector $x \in \mathcal{L} \setminus 0$ such that

$$\|x\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L}),$$

where $\lambda_1(\mathcal{L})$ denotes the length of the shortest vector in the lattice.

Definition 2.3 (γ -CVP [SD16]). Given a lattice $\mathcal{L} \subset \mathbb{R}^n$ and a target vector $t \in \mathbb{R}^n$, find a lattice vector $x \in \mathcal{L}$ such that

$$\|x - t\| \leq \gamma(n) \cdot \text{dist}(\mathcal{L}, t),$$

where $\text{dist}(\mathcal{L}, t) = \min_{x \in \mathcal{L}} \|x - t\|$. See Figure 2 for a diagram.

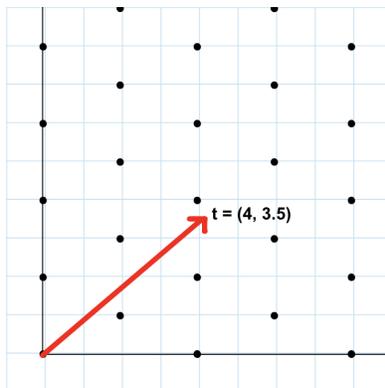


Figure 2: An instance of the CVP, with target vector $t = (4, 3.5)$ and solution $(4, 4)$. Note that the shortest vector in this lattice is $(0, 2)$.

These problems have many variants. In addition to the approximation versions presented above, both γ -SVP and γ -CVP have exact analogues (called SVP and CVP respectively), where the goal is to find the shortest/closest vector instead of approximating it. And, γ -SVP also has a decision-based version called γ -GAP-SVP:

Definition 2.4 (γ -GAP-SVP [Reg10]). Given a lattice $\mathcal{L} \subset \mathbb{R}^n$ and $d > 0$, output YES if $\lambda_1(\mathcal{L}) \leq d$ and NO if $\lambda_1(\mathcal{L}) \geq \gamma(n) \cdot d$.

A slightly different problem, considered by Eldar and Hallgren [Hal21], is Bounded Distance Decoding (BDD), which is concerned with recovering a lattice vector given a target vector with a slight error.

Definition 2.5 (Bounded Distance Decoding [DvW21b]). Given a lattice $\mathcal{L} \subset \mathbb{R}^n$, a radius $\frac{\lambda_1(\mathcal{L})}{2} > r > 0$, and a vector $t = v + e$ with $\|e\| < r$ and $v \in \mathcal{L}$, output the lattice vector v .

Note that the condition $r < \frac{\lambda_1(\mathcal{L})}{2}$ guarantees a unique solution. BDD is the main problem that we will be studying in this paper, since both the quantum and classical algorithms discussed below give subexponential approximations. As we will develop in Section 4, there are classical polynomial-time algorithms that can approximate SVP and CVP for $\gamma(n) = 2^{O(n)}$; namely, the LLL algorithm and Babai’s nearest plane algorithm. More concretely, the LLL algorithm solves γ -SVP for $\gamma(n) = 2^{(n-1)/2}$, whereas Babai’s nearest plane algorithm solves γ -CVP for $\gamma(n) = 2^{n/2}$.¹ One can then use the same algorithm by Babai to solve BDD whenever $2^{n/2} \cdot \text{dist}(\mathcal{L}, t) \leq \lambda_1/2$, given that in that case the answer is unique.

2.3 Lattice Hardness Results

There are several important intractability results about such computational lattice problems. In 1981, van Emde Boas proved that CVP is NP-hard, but the same result did not immediately extend to SVP [vEB81]. In fact, γ -CVP is now known to be NP-complete for any $\gamma \leq n^{c/\log \log n}$. In 1997, Ajtai proved that SVP is NP-hard to solve exactly under randomized reduction [Ajt98], which provided the first theoretical evidence that SVP is intractable. Later on, Micciancio proved that SVP is NP-hard to approximate with any factor less than $\sqrt{2}$ [Mic01]. Lastly, Goldreich, Micciancio, Safra, and Seifer showed that SVP is not harder than CVP [GMSS99]. Additionally, the various lattice problems are connected by a network of reductions. Figure 3 contains an overview of the major dimension-preserving reductions.

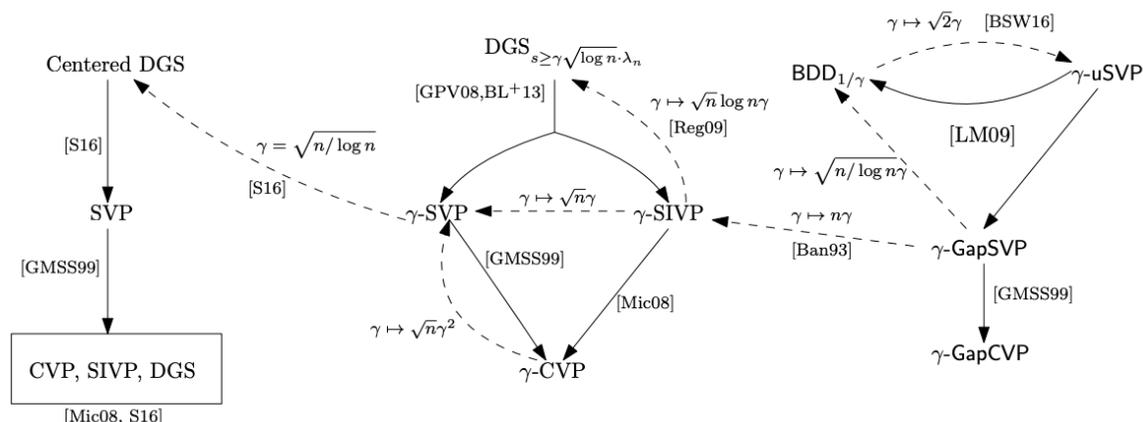


Figure 3: A summary of significant dimension-preserving reductions between various lattice problems, where $A \rightarrow B$ means A reduces to B . Solid (dotted) arrows denote a reduction that preserves (changes) the approximation factor γ . Source: [SD21].

While interesting in their own right, lattice problems are especially relevant for cryptography. Currently, the best candidates for post-quantum cryptography are lattice-based encryptions, and the security of these schemes relies on the conjectured hardness of these lattice problems. When Learning With Errors (LWE) was first introduced by Regev he proved its security by reducing it to γ -GAP-SVP:

Theorem 2.6 (Regev, Hardness of Learning With Errors [Reg10]). *Define $LWE_{p,\chi}$ as follows: given a series of equations of the form*

$$\langle s, a_i \rangle + e_i = b_i$$

where $s \in \mathbb{Z}_p^n$, the a_i are independently and uniformly chosen from \mathbb{Z}_p^n , $b_i \in \mathbb{Z}_p$, and the e_i are independently chosen from χ , recover s . Then, if $\alpha \in (0, 1)$ is such that $\alpha p > 2\sqrt{n}$, and there is an

¹By tuning the parameters of the algorithm, it is actually possible to improve the approximation factors to $\gamma(n) = (2/\sqrt{3})^n$ and $2(2/\sqrt{3})^n$, respectively [Reg04b]. This approximation factor can be further improved with BKZ bases [LLL82].

efficient algorithm that can solve $LWE_{p, \bar{\Psi}_\alpha}$, then there exists an efficient quantum algorithm that solves $\tilde{O}(n/\alpha)$ -GAP-SVP in the worst case.

The distribution $\bar{\Psi}_\alpha$ is described at depth in the paper, but it is essentially a discrete Gaussian centered at 0 with standard deviation αp .

Crucially, γ -GAP-SVP is conjectured to be hard to solve, i.e., no polynomial time algorithm can solve it for a polynomial γ . And, Micciancio proved that for $\gamma(n) \leq \sqrt{2}$, γ -GAP-SVP is NP-Hard [Mic01]. Though Regev’s reduction is quantum, Peikert demonstrated a classical reduction in [Pei09], and in 2009, Lyubashevsky and Micciancio [LM09] reduced γ -GAP-SVP to BDD:

Theorem 2.7 (Theorem 7.1 in [LM09]). *For any $\gamma > 2\sqrt{n/\log n}$, there is a polynomial time Cook reduction from γ -GAP-SVP to $\frac{1}{\gamma}\sqrt{n/\log n}$ -BDD.*

Elsewhere in the paper, they proved the opposite direction of the reduction as well, meaning that the conjectured hardness of γ -GAP-SVP is equivalent to the hardness of BDD. It is precisely because of this link between the security of LWE and BDD that further study of Eldar and Hallgren’s algorithm for BDD is so important.

2.4 Survey of Quantum Approaches to Lattice Problems

Before Eldar and Hallgren’s algorithm for BDD, a number of other quantum algorithms have been proposed for different lattice problems. For instance, Kuperberg describes in [Kup05] a quantum algorithm utilizing Quantum Fourier Transform (QFT) to solve the dihedral hidden subgroup problem (DHSP) that has time complexity $2^{O(\sqrt{\log N})}$ and takes $2^{O(\sqrt{\log N})}$ quantum space, where $2N$ is the order of the dihedral group D_N . Regev applies a similar quantum algorithm to the unique shortest vector problem (unique SVP) in [Reg04c] and proposes a space-efficient alternative to Kuperberg’s algorithm in [Reg04d], taking only $O(\log N)$ quantum space instead of $2^{O(\sqrt{\log N})}$. Kuperberg then generalizes Regev’s algorithm in [Kup11], adding two parameters that (1) allows the algorithm to take less space at the cost of more quantum time and (2) allows the algorithm take less quantum time at the cost of more classical space and time, as long as the the classical space has quantum access. However, the composition of Kuperberg and Regev’s algorithm does *not* yield a subexponential algorithm for SVP. The reason is that Regev’s reduction produces a quadratic blowup in the input size.

Overall, these papers are good examples of how quantum algorithms can be used in order to improve classical approaches to lattice-based problems by providing time and space efficient variations, which we imagine was also what motivated Eldar and Hallgren’s work.

3 Overview of Eldar and Hallgren’s Algorithm

We now give an overview of some of the key points in Eldar and Hallgren’s proof of Theorem 1.1 [EH22]. Eldar and Hallgren parametrize lattices by the following three integers:

1. Lattice dimension n : n is the dimension of the space \mathbb{Z}^n in which our lattice is embedded. We consider only full rank lattices, i.e., those for which the cardinality of the basis \mathbf{B} is also n , since any lattice can be reduced to a full-rank lattice in a lower-dimensional \mathbb{Z}^m .
2. Periodicity q : q is defined as the minimal integer such that $q\mathbb{Z}^n \subset \mathcal{L}$. In other words, it is the minimal integer such that $qe_i \in \mathcal{L}$ for every standard basis vector $e_i \in \mathbb{Z}^n$. Such an integer must exist. To see why, note that there must be some minimal integer q_i such that $q_i e_i \in \mathcal{L}$, because otherwise we could extend our basis \mathbf{B} by e_i , which is impossible since \mathcal{L} is full-rank. Then let $q = \text{LCM}(\{q_i\})$.
3. Finite group rank k : Since $q\mathbb{Z}^n$ is a subgroup of the abelian group \mathcal{L} , we can consider the quotient $\mathcal{L} \bmod q \equiv \mathcal{L}/q\mathbb{Z}^n$, obtained by taking each standard basis coefficient of a lattice vector mod q . This is a finite abelian group since \mathbb{Z}_q is finite, so by the structure theorem for finitely generated abelian groups we can decompose $\mathcal{L} \bmod q$ as $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_k}$ for $q_i \mid q_{i+1}$. Moreover, k is the group rank of $\mathcal{L} \bmod q$ and can be thought of as the dimension of the reduced lattice, analogous to the security parameter in LWE. Letting \mathbf{G} give a basis for $\mathcal{L} \bmod q$, it

follows that the matrix $[\mathbf{G}|q\mathbf{I}]$ for an $n \times n$ identity \mathbf{I} generates the full lattice \mathcal{L} (but is not a basis; it is too large to be a linearly independent set).

Now that we understand the claim, we begin with a review of some major quantum primitives used in the construction.

3.1 Quantum Primitives

There are only two quantum primitives necessary to understand Eldar and Hallgren’s algorithm, and one is derived from the other. First is the Quantum Fourier Transform (QFT), a particular basis change for quantum states.

Definition 3.1. Let $|x\rangle$ for $0 \leq x < q$ denote a basis state for some q -dimensional Hilbert space \mathcal{H} of quantum states. The Quantum Fourier Transform (QFT) over the cyclic group \mathbb{Z}_q , denoted F_q , is a linear map defined by

$$F_q : |x\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{i=0}^{q-1} \omega_q^{ix} |i\rangle,$$

where ω_q denotes a primitive q th root of unity.

The QFT is computable in $\text{poly}(\log(q))$ time using only Hadamard and controlled phase rotations. Note that this definition differs from that presented in class and [Bar22], which uses only Hadamard gates and no controlled operations, but conforms to the definition most often used in physics (by leaving out the controlled phase rotations we obtain what physicists refer to as the *Hadamard transformation*). One of the main uses of the QFT is as the basic building block for the Quantum Phase Estimation (QPE) algorithm:

Theorem 3.2 (Pg. 225 in [NC10]). *There exists a quantum algorithm which, given a unitary U , and an eigenstate $|u\rangle$ of U with eigenvalue $e^{2\pi i\phi_u}$, outputs an n -bit approximation $\tilde{\phi}_u$ to ϕ_u with probability $1 - \epsilon$ using $t = n + O(\log(1/\epsilon))$ ancilla qubits and $O(t^2)$ operations.*

The algorithm works by first applying a series of controlled unitary powers to the ancilla qubits, which transforms our ancilla register into a QFT state. We can then apply the inverse QFT to obtain an approximation of the phase of the unitary eigenvalue.

With these quantum primitives covered, we can understand the basic arguments behind Eldar and Hallgren’s algorithm, presented below.

3.2 Intuition for the Proof

Eldar and Hallgren’s algorithm reduces worst-case BDD on a lattice \mathcal{L} in $\mathcal{L}(n, q, k)$ to $\tilde{\epsilon}$ -random-BDD, which is BDD on a random lattice $\tilde{\mathcal{L}}$ with basis $[\tilde{\mathbf{B}}|q\mathbf{I}]$ for $\tilde{\mathbf{B}} \in \mathbb{Z}_q^{m \times k}$ selected randomly. $\tilde{\mathcal{L}}$ has shortest vector of length $\tilde{\lambda}_1$, and the transformed target vector \tilde{t} is at most $\tilde{\lambda}_1 \tilde{\epsilon}$ away from a lattice vector in $\tilde{\mathcal{L}}$. For $m = \sqrt{k \log q}$, we can solve BDD on this lattice with approximation factor $\tilde{\epsilon} = 2^{-\sqrt{k \log q}}$ using Babai’s nearest plane algorithm.

Before obtaining this random lattice, we consider a related, long-standing problem in quantum state representations of a lattice. For decades, researchers have considered the goal of generating, given a lattice \mathcal{L} with coefficients in \mathbb{Z}_q with basis \mathbf{B} , the state $|\psi_0\rangle = \sum_{v \in \mathcal{L}} \sum_{z \in C} |v + z\rangle$ where C is some extended shape, say a cube (more on why this is desirable below). We approximate $|\psi_0\rangle$ in the following way. Begin by computing the state $\sum_c |c\rangle \otimes \sum_{z \in C} |z\rangle$ where c ranges over all possible coefficients of lattice vectors in \mathcal{L} . We can then perform entangling gates to obtain $\sum_c \sum_{z \in C} |c, \mathbf{B}c + z\rangle$. We want the state $\sum_c \sum_{z \in C} |0, \mathbf{B}c + z\rangle$, but computing this would require us to know the c corresponding to each $\mathbf{B}c + z$, i.e., to solve BDD, which is exactly the problem we are trying to solve in the first place!

We cannot do this, but we can perform a Quantum Fourier Transform and measure the first register to compute $|\psi_a\rangle = \sum_c \omega_q^{c \cdot a} \sum_{z \in C} |\mathbf{B}c + z\rangle$ for a a random vector (random by the nature of quantum measurement). The problem of computing c from $\mathbf{B}c + z$ is therefore reduced to the problem of determining the phases $\omega_q^{c \cdot a}$. The key observation is that the state $|\psi_a\rangle$ is an eigenvector of a shift by a lattice vector. For $x \in \mathbb{Z}_q^n$, let U_x be the shift unitary defined by $U_x |y\rangle = |y + x\rangle$.

Then if we shift by the closest lattice vector $\mathbf{B}c' \in L$ to t , we obtain $U_{Bc'} |\psi_a\rangle = \omega_q^{c' \cdot a} |\psi_a\rangle$. Applying the Quantum Phase Estimation (QPE) algorithm, we can compute $c' \cdot a$, and after repeating for multiple random a we can obtain a good approximation of c' .

Unfortunately, we are trying to solve for $\mathbf{B}c'$, so we cannot proceed so directly. However, we do have the target vector $t = \mathbf{B}c' + \Delta$ for some Δ of norm regulated by the BDD approximation factor, and we have $U_t |\psi_a\rangle = \omega_q^{c' \cdot a} U_\Delta |\psi_a\rangle \approx \omega_q^{c' \cdot a} |\psi_a\rangle$. (“Approximately” here is defined in terms of state overlap. If we had not summed over cubes C about each lattice point, then for any Δ not a lattice vector this overlap would be zero. This justifies our consideration of these states). This is the key quantum subroutine used in the dimensionality reduction: Eldar and Hallgren use a modified approximate QPE to sample $m = O(\sqrt{k \log q})$ noisy inner products of c' and construct a lattice $\tilde{\mathcal{L}}$ of the form described above. Then they simply use Babai’s nearest plane algorithm on $\tilde{\mathcal{L}}$ and map the solution back to the original lattice \mathcal{L} .

4 Classical Primitives

4.1 The LLL Algorithm

These lattice problems had been studied long before they became relevant to cryptography after Ajtai’s worst-case/average-case equivalence result for lattices [Ajt98]. In fact, Gauss already provided a polynomial-time algorithm to solve SVP *exactly* in 2-dimensions in the 19th century. Broadly, the algorithm finds the most orthogonal basis we can find for the given lattice using orthogonal projections, in an iterative manner that resembles Euclid’s algorithm [Gal12].

While solving SVP exactly (i.e., for $\gamma(n) = 1$) does not carry over to higher dimensions, a similar idea prevails. Namely, we would like to obtain a short basis, which is achieved through orthogonalizing such basis as much as possible. Suppose a basis $\{v_1, \dots, v_n\}$ is orthogonal. The shortest vector in the lattice can be written as some linear combination $\lambda_1 = a_1 v_1 + \dots + a_n v_n$ for $a_i \in \mathbb{Z}$, which has norm-squared $\|\lambda_1\|^2 = a_1^2 \|v_1\|^2 + \dots + a_n^2 \|v_n\|^2$. This is minimized for $a_i = \delta_{j0}$, such that v_j is the shortest basis vector. We conclude that any orthogonal basis must include the shortest vector in the lattice. While not every lattice has an orthogonal basis, this reasoning explains why the LLL algorithm uses the Gram-Schmidt orthogonalization process as a subroutine. Throughout this paper, we will use the superscript $*$ to denote a Gram-Schmidt vector.

The LLL algorithm was introduced by Arjen Lenstra, Hendrik Lenstra, and László Lovász in 1982 [LLL82], originally in the context of polynomial factorization. The LLL algorithm reduces an arbitrary basis for a lattice into a “shorter” basis, which is referred to as the *LLL-reduced basis* and is formally defined as follows:

Definition 4.1 (δ -LLL reduced basis). Let $\{b_1, b_2, \dots, b_n\}$ be a basis for an n -dimensional lattice \mathcal{L} , and let $\{b_1^*, b_2^*, \dots, b_n^*\}$ be the orthogonal basis generated with Gram-Schmidt. Let $\mu_{i,k} = \frac{\langle b_k, b_i^* \rangle}{\langle b_i^*, b_i^* \rangle}$, where $\langle \cdot, \cdot \rangle$ denotes the dot product between two vectors. We say that $\{b_1, b_2, \dots, b_n\}$ is a δ -LLL-reduced basis if it satisfies the following two conditions:

1. (Size reduced.) For all $i \neq k, \mu_{i,k} \leq 1/2$.
2. (Lovász condition.) For each $i, \|b_{i+1}^* + \mu_{i,i} b_i^*\|^2 \geq \delta \|b_i^*\|^2$. Equivalently, we can write $\|b_{i+1}^*\|^2 \geq (\delta - \mu_{i+1,i}^2) \|b_i^*\|^2$.

We remark that, in [DvW21b], their usages of δ denote what we have defined as $1/\sqrt{\delta}$. That is, if δ' denotes the parameter used by [DvW21b], then $\delta' = 1/\sqrt{\delta}$, where δ is defined as above. It is customary in the literature to choose $\delta = 3/4$, although any value $\delta \in (1/4, 1)$ guarantees that the LLL algorithm terminates in polynomial time.

Put into words, the second condition reads as saying that b_i^* is not much larger than b_{i-1}^* . Let us further develop the Lovász condition:

$$\|b_{i+1}^*\|^2 \geq (\delta - \mu_{i+1,i}^2) \|b_i^*\|^2 \geq \delta \|b_i^*\|^2,$$

$$\|b_i^*\| \geq \sqrt{\delta} \|b_{i-1}^*\|.$$

If we take logarithms in the Lovász condition, we obtain that

$$\ln \|b_{i+1}^*\| \geq \ln(\sqrt{\delta} \cdot \|b_i^*\|) = \ln \|b_i^*\| - \ln(1/\sqrt{\delta}). \quad (4.1)$$

In notation of the Ducas and van Woerden note, where $\ell_i := \ln \|b_i^*\|$, this corresponds to

$$\ell_{i+1} \geq \ell_i - \ln(1/\sqrt{\delta}). \quad (4.2)$$

If we re-state Equation 4.2 in terms of the δ' used in the Ducas and van Woerden note, we obtain that

$$\ell_{i+1} \geq \ell_i - \ln(\delta'). \quad (4.3)$$

This corresponds exactly to Fact 2.2 in the Ducas and van Woerden note.

It is natural to ask whether every lattice has an LLL-reduced basis. The LLL algorithm gives a constructive proof that the answer is yes. The LLL pseudocode is presented in Algorithm 1.

Algorithm 1 The LLL algorithm

```

1: procedure LLLALG( $\{b_1, b_2, \dots, b_n\}$ )
2:   Compute the Gram-Schmidt basis  $b_1^*, \dots, b_n^*$  and coefficients  $\mu_{i,j}$  for  $1 \leq j < i \leq n$ .
3:   Repeat the following two steps until the LLL reduced basis is found:
4:   for  $i = 1$  to  $n$  do ▷ Perform size reduction
5:     for  $k = i - 1$  to  $1$  do
6:        $m \leftarrow \lfloor \mu_{k,i} \rfloor$ 
7:        $b_i \leftarrow b_i - mb_k$ 
8:     end for
9:   end for
10:  for  $i = 1$  to  $n - 1$  do ▷ Check Lovász condition
11:    if  $\|b_{i+1}^* + \mu_{i,i+1}b_i^*\|^2 < \delta \|b_i^*\|^2$  then
12:      Swap  $b_{i+1}$  and  $b_i$ .
13:      Go to Step 1.
14:    end if
15:  end for
16: end procedure

```

The correctness of the algorithm can be stated as follows [Gal12]:

Lemma 4.2. *If the LLL algorithm terminates, then the output basis is LLL-reduced. Moreover, for any $\delta \in (1/4, 1)$, the LLL algorithm terminates within polynomial time.*

We will focus on the proof that the LLL algorithm terminates, because such a proof uses facts that we will need when showing correctness of the classical algorithm in Ducas and van Woerden. To show that the algorithm performs a finite number of steps, we define the following *potential function* on the basis vectors of the lattice [Reg04a]:

$$\phi(\mathbf{B}) = \prod_{i=1}^n \det(\mathcal{L}_i) = \prod_{i=1}^n \|b_1^*\| \cdot \dots \cdot \|b_i^*\| = \prod_{i=1}^n \|b_i^*\|^{n-i+1}, \quad (4.4)$$

where \mathcal{L}_i is defined as the lattice spanned by b_1, \dots, b_i . Then, to show that the LLL algorithm terminates, we need to prove that the initial value of $\phi(\mathbf{B})$ is upper bounded, that it can never drop below 1, and that each iteration of the LLL algorithm decreases ϕ by a factor of at least $\sqrt{4/3}$ [Pei13].

The potential of the original basis \mathbf{B} is bounded by $\prod_{i=1}^n \|b_i\|^n$, given that $\|b_i^*\| \leq \|b_i\|$. Because every intermediate basis is integral by definition and has positive integer determinant, the same holds for all the lattices \mathcal{L}_i associated with that basis. Hence it follows that the potential is at least 1.

Let us consider the step *Check Lovász condition* in the pseudocode of the LLL algorithm (namely, Line 10). Suppose that b_i and b_{i+1} are swapped in Line 12 of the LLL algorithm pseudocode, and let the resulting basis be denoted \mathbf{B}' . Then, we claim the following [Pei13]:

Lemma 4.3. *If b_i and b_{i+1} are swapped in Line 12 of the LLL algorithm, then $b_j^* = b_j^*$ for all $j \notin \{i, i+1\}$, and $b_i^* = \mu_{i,i+1} b_i^* + b_{i+1}^*$.*

Proof. If $j < i$, then the vector b_j^* is not affected by the swap, because it is the component of $b_j' = b_j$ orthogonal to $\text{span}(b_1', \dots, b_{j-1}') = \text{span}(b_1, \dots, b_{j-1})$. The same argument holds for $j > i+1$. Lastly, b_i^* is by definition the component of $b_i' = b_{i+1}$ orthogonal to $\text{span}(b_1', \dots, b_{i+1}') = \text{span}(b_1, \dots, b_{i-1})$, which is $\mu_{i,i+1} b_i^* + b_{i+1}^*$ by construction. \square

We can then prove the following [Pei13]:

Lemma 4.4. *If b_i and b_{i+1} are swapped in Line 12 of the LLL algorithm, then*

$$\frac{\phi(\mathbf{B}')}{\phi(\mathbf{B})} < \sqrt{\delta}.$$

Proof. This immediately follows from the previous lemma. Define \mathcal{L}_i as before, and let \mathcal{L}'_i denote the lattice generated by the first i basis vectors after a swap. Then:

$$\frac{\phi(\mathbf{B}')}{\phi(\mathbf{B})} = \frac{\det(\mathcal{L}'_i)}{\det(\mathcal{L}_i)} = \frac{\|b_1^*\| \cdot \dots \cdot \|b_{i-1}^*\| \cdot \|\mu_{i,i+1} b_i^* + b_{i+1}^*\|}{\|b_1^*\| \cdot \dots \cdot \|b_{i-1}^*\| \cdot \|b_i^*\|} = \frac{\|\mu_{i,i+1} b_i^* + b_{i+1}^*\|}{\|b_i^*\|} < \sqrt{\delta},$$

given that, as per Line 11 in the LLL algorithm (Alg. 1), the vectors are swapped if and only if $\|b_{i+1}^* + \mu_{i,i+1} b_i^*\|^2 < \delta \|b_i^*\|^2$. \square

Since $\delta \in (1/4, 1)$, the previous lemma in particular implies that

$$\prod_{j=i}^n \det(\mathcal{L}'_j) \leq \prod_{j=i}^n \det(\mathcal{L}_j).$$

By taking logarithms, and by switching to the notation of Ducas and van Woerden, it follows that

$$\sum_{j=1}^i \ell_j \leq \sum_{j=1}^i \ell'_j, \tag{4.5}$$

which corresponds to their Fact 2.3. Note that in Ducas and van Woerden's notation, ℓ'_j is the logarithm of the norm of a vector from the original basis, and ℓ_j is the logarithm of the norm of a vector from the LLL-reduced basis.

Lastly, we ask, how does the LLL-reduced basis relate to the SVP? The following theorem shows their relationship; namely, the first vector in the ordered LLL-reduced basis corresponds to the desired approximation of $\lambda_1(\mathcal{L})$:

Proposition 4.5 (γ -SVP from LLL). *If $\{b_1, b_2, \dots, b_n\}$ is an n -dimensional LLL-reduced basis of a lattice \mathcal{L} , then $\|b_1\| \leq 2^{(n-1)/2} \lambda_1(\mathcal{L})$, where $\lambda_1(\mathcal{L})$ is the length of the shortest vector of \mathcal{L} .*

The proof is short and can be consulted, for example, in Deng's manuscript [Den16].

4.2 Babai's Nearest Plane Algorithm

After explaining how the LLL algorithm provides an approximation to SVP, let us mention briefly how to obtain an approximation to CVP.

Babai's nearest plane algorithm solves γ -CVP for $\gamma(n) = 2^{n/2}$ in polynomial time. Because we do not need the inner mechanisms of Babai's algorithm in order to analyze the classical counterpart to Eldar and Hallgren's algorithm, other than its final approximation factor for CVP, we will not describe the algorithm in detail and instead only provide a short intuitive overview.

Babai’s algorithm considers the affine subspace $H_c = \{\sum_{i=1}^{n-1} a_i b_i + c b_n\}$. Then, $H_c = H_0 + c b_n$. However, b_n is not necessarily orthogonal to H_0 , and hence the distance between H_0 and H_c is exactly $c \|b_n^*\|$ (as opposed to $c \|b_n\|$). Babai then observes that inside each of these hyperplanes, there is a copy of the sublattice \mathcal{L}' generated by the first $n - 1$ basis vectors b_i and shifted by $c b_n$. Then, Babai’s algorithm iteratively finds the closest hyperplane H_c to the target vector t (as defined in γ -CVP) fixing one coordinate at a time [Bab86].

When the input lattice to Babai’s algorithm is an LLL-reduced basis (as described in Section 4.1), the desired approximation for CVP follows. A proof of this claim can be found, for example, in [SD16] or [Reg04b].

5 Proof of Classical Alternative

Eldar and Hallgren’s proof is interesting and uses “phased cube states” in a unique way, but their construction can be simplified. For the class of lattices on which they solve BDD, there in fact exists a deterministic, far simpler classical algorithm achieving the same result. Within three days of Hallgren’s talk [Hal21], Ducas and van Woerden published a note [DvW21a] solving BDD with the appropriate subexponential factor for the special case $q = 2^n, k = 1$.

Emboldened by this result, we sought to generalize their proof to the case of arbitrary q, k , demonstrating a classical alternative to Eldar and Hallgren’s algorithm in all cases. However, on October 14, 2021, Ducas and van Woerden managed to prove the generalization we were working on [DvW21b]. Still, their proof is rather brief and leaves out many key details and intuitions. Here, we present a complete and self-contained proof of Theorem 1.3.

We remark that we already proved Facts 2.2 and 2.3 in [DvW21b] when describing the LLL algorithm and its correctness proof in Section 4.1. We will use them again when finalizing the proof of the result in Section 5.3.

5.1 Short Bases from Short Full-Rank Sets

We open the proof with a lemma which shows that if we can find a full-rank set of short lattice vectors, then we can construct a lattice basis of short vectors, as well. Formally:

Lemma 5.1 (Lemma 7.1 in [MG02], Fact 2.1 in [DvW21b]). *There exists a deterministic poly-time algorithm which, given a basis \mathbf{B}' of an n -dimensional lattice $\mathcal{L}(\mathbf{B}')$, and a full-rank set of n vectors $S \subset \mathcal{L}(\mathbf{B}')$ outputs a basis \mathbf{B} such that $\|b_i^*\| \leq \max_{s \in S} \|s\|$.*

We will make use of two subclaims regarding the relationships between unimodular and upper-triangular matrices. Unimodular matrices are defined below:

Definition 5.2. An integer matrix $U \in \mathbb{Z}^{n \times n}$ is *unimodular* if $\det(U) = \pm 1$. Note that any unimodular matrix has integer inverse by Cramer’s Theorem.

We prove the following lemma:

Lemma 5.3. *Given an integer matrix $Q \in \mathbb{Z}^{n \times n}$, there exists a unimodular matrix $U \in \mathbb{Z}^{n \times n}$ such that $T = UQ$ is upper-triangular.*

Proof. U will be constructed as a product of unimodular matrices corresponding to elementary matrices for row operations on an integer matrix: swapping two rows, multiplying a row by -1 , and adding an integer multiple of one row to another. All such elementary matrices have determinant ± 1 , so by the homomorphism property of the determinant, so will U . We proceed by induction: if we can show that given a list $\{a_1, \dots, a_k\}$ of integers (which we think of as the non-zero entries of a column of some U' obtained from U via a preceding sequence of elementary row operations), we can produce a new list $\{0, a'_1, \dots, a'_{k-1}\}$ of integers by swapping entries in the list, multiplying entries by -1 , and adding an integer multiple of one entry to another, then the proof will follow. First, multiply the appropriate entries by -1 such that all entries are non-negative. Then, order the entries by increasing value. If the first entry is 0, we are done; otherwise, subtract the maximal integer multiple of the first entry from all other entries which preserves non-negativity, and reorder in increasing value. The new first entry must have decreased in value from the previous first entry.

This procedure is guaranteed to terminate in a finite number of steps, such that the first entry in eventually zero, which completes the proof. \square

In addition, we will need the following lemma:

Lemma 5.4. *Suppose $S = BT$ for T upper-triangular and invertible. Then $s_i^* = T_{ii}b_i^*$ for s_j, b_j the j th columns of S and B respectively.*

Proof. We proceed by induction. Suppose the claim holds for all $j \in [i - 1]$. Then we have

$$s_i^* = s_i - \sum_{j=1}^{i-1} \frac{\langle s_j^*, s_i \rangle}{\langle s_j^*, s_j^* \rangle} s_j^* = s_i - \sum_{j=1}^{i-1} \frac{\langle b_j^*, s_i \rangle}{\langle b_j^*, b_j^* \rangle} b_j^*.$$

Note that s_i can be written as $s_i = \sum_{k=1}^i T_{ki}b_k$ since T is upper-triangular, so

$$s_i^* = \sum_{k=1}^i T_{ki}b_k - \sum_{j=1}^{i-1} \frac{\langle b_j^*, \sum_{k=1}^i T_{ki}b_k \rangle}{\langle b_j^*, b_j^* \rangle} b_j^* = \sum_{k=1}^i T_{ki}b_k - \sum_{k=1}^i \sum_{j=1}^{i-1} \frac{\langle b_j^*, T_{ki}b_k \rangle}{\langle b_j^*, b_j^* \rangle} b_j^*.$$

Consider just the term

$$\sum_{k=1}^{i-1} \sum_{j=1}^{i-1} \frac{\langle b_j^*, T_{ki}b_k \rangle}{\langle b_j^*, b_j^* \rangle} b_j^*.$$

Since $b_k \in \text{span}\{b_1^*, \dots, b_{i-1}^*\}$ for all $k \in [i - 1]$, it follows that

$$\sum_{j=1}^{i-1} \frac{\langle b_j^*, T_{ki}b_k \rangle}{\langle b_j^*, b_j^* \rangle} b_j^* = T_{ki}b_k.$$

Therefore, we have

$$s_i^* = T_{ii}b_i - T_{ii} \sum_{j=1}^{i-1} \frac{\langle b_j^*, b_i \rangle}{\langle b_j^*, b_j^* \rangle} b_j^* \equiv T_{ii}b_i^*,$$

as desired. \square

Armed with these intermediate results, we may now return to the proof of the main lemma:

Proof. Since S is a set of lattice vectors, we can write $S = \mathbf{B}'Q$ for some integer matrix $Q \in \mathbb{Z}^{n \times n}$, which is invertible since S is full rank. By Lemma 5.3, we may write $T = UQ$ for some unimodular U such that T is upper-triangular. Let $\mathbf{B} = \mathbf{B}'U^{-1}$. \mathbf{B} is a basis since U is unimodular, and therefore U^{-1} is an integer matrix. Furthermore, $S = \mathbf{B}T$, so by Lemma 5.4 we have $\|b_i^*\| = \frac{1}{|T_{ii}|} \|s_i^*\| \leq \|s_i^*\|$, since T_{ii} are all integers. Since $\|s_i^*\| \leq \|s_i\| \leq \max_{s \in S} \|s\|$, the proof is complete. \square

5.2 Duality

In order to complete our proof, we will be using the idea of dual lattices, the definition of which is given as:

Definition 5.5. The dual of a lattice \mathcal{L} is

$$\hat{\mathcal{L}} = \{v \in \text{span}(\mathcal{L}) \mid \langle v, w \rangle \in \mathbb{Z} \text{ for all } w \in \mathcal{L}\}. \quad (5.1)$$

Equivalently, the dual lattice $\hat{\mathcal{L}}$ can be thought of as the set of all linear functions $\mathcal{L} \rightarrow \mathbb{Z}$. In order to use duality in our proof, we will first need to prove several lemmas, some of which are inspired by the exercises in [Mic07] by Daniele Micciancio.

Lemma 5.6. *The dual of \mathbb{Z}^n is \mathbb{Z}^n .*

Proof. Take any $v \in \mathbb{Z}^n$. Since integers are closed under addition and multiplication, we have $\langle v, w \rangle \in \mathbb{Z}$ for any $w \in \mathbb{Z}^n$. This shows $\mathbb{Z}^n \subset \hat{\mathbb{Z}}^n$. Similarly, take any $v \in \hat{\mathbb{Z}}^n$. For any $i \in \{1, \dots, n\}$ we have $e_i \in \mathbb{Z}^n$ (where e_i is the unit vector with 1 for the i th coordinate and 0 for the remaining coordinates). By definition of the dual lattice, this implies $\langle v, e_i \rangle = v_i \in \mathbb{Z}$. Hence, $v \in \mathbb{Z}^n$ and $\hat{\mathbb{Z}}^n \subset \mathbb{Z}^n$. This proves that the dual of \mathbb{Z}^n is \mathbb{Z}^n . \square

Lemma 5.7. *Let $\hat{\mathcal{L}}$ be the dual lattice of \mathcal{L} . For any $d > 0$, the dual lattice of $d\mathcal{L}$ is $\frac{1}{d}\hat{\mathcal{L}}$.*

Proof. Take any $v \in (d\hat{\mathcal{L}})$. For all $w \in \mathcal{L}$, we have $dw \in d\mathcal{L}$, and hence:

$$\langle v, dw \rangle \in \mathbb{Z} \Rightarrow \langle dv, w \rangle \in \mathbb{Z} \Rightarrow dv \in \hat{\mathcal{L}} \Rightarrow v \in \frac{1}{d}\hat{\mathcal{L}} \Rightarrow (d\hat{\mathcal{L}}) \subset \frac{1}{d}\hat{\mathcal{L}}. \quad (5.2)$$

Similarly, take any $v \in \frac{1}{d}\hat{\mathcal{L}}$, implying $dv \in \hat{\mathcal{L}}$. Note that for all $w \in d\mathcal{L}$, we have $\frac{w}{d} \in \mathcal{L}$. Hence:

$$\frac{w}{d} \in \mathcal{L}, dv \in \hat{\mathcal{L}} \Rightarrow \left\langle dv, \frac{w}{d} \right\rangle \in \mathbb{Z} \Rightarrow \langle v, w \rangle \in \mathbb{Z} \Rightarrow v \in (d\hat{\mathcal{L}}) \Rightarrow \frac{1}{d}\hat{\mathcal{L}} \subset (d\hat{\mathcal{L}}) \quad (5.3)$$

Hence, $\frac{1}{d}\hat{\mathcal{L}}$ is the dual of $d\mathcal{L}$. \square

Theorem 5.8 (Theorem 2 from [Mic07]). *The dual of a lattice with basis \mathbf{B} is a lattice with basis (the dual basis) $\mathbf{D} = \mathbf{B}\mathbf{G}^{-1}$, where $\mathbf{G} = \mathbf{B}^\top \mathbf{B}$ is the Gram matrix of \mathbf{B} .*

Proof. For any basis \mathbf{B} , define \mathbf{D} as given in the theorem. The invertibility of \mathbf{G} follows from the linear independence of the columns of \mathbf{B} (hence $\mathbf{B}^\top \mathbf{B}$ is nonsingular). Say $\mathcal{L}(\mathbf{B})$ and $\mathcal{L}(\mathbf{D})$ are the lattices corresponding to bases \mathbf{B} and \mathbf{D} , respectively, and $\hat{\mathcal{L}}(\mathbf{B})$ and $\hat{\mathcal{L}}(\mathbf{D})$ are their dual lattices. From Definition 2.1, it follows that any vector in $\mathcal{L}(\mathbf{D})$ is of form $\mathbf{D}y$ for some integer vector y . Accordingly, take any $\mathbf{D}y \in \mathcal{L}(\mathbf{D})$. We have:

- $\mathbf{D}y = \mathbf{B}\mathbf{G}^{-1}y = \mathbf{B}((\mathbf{B}^\top \mathbf{B})^{-1}y) \in \text{span}(\mathbf{B})$.
- For all $\mathbf{B}x \in \mathcal{L}(\mathbf{B})$, we have $\langle \mathbf{D}y, \mathbf{B}x \rangle = (\mathbf{B}x)^\top (\mathbf{D}y) = x^\top \mathbf{B}^\top \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1}y = x^\top y \in \mathbb{Z}$.

Hence, we have $\mathbf{D}y \in \hat{\mathcal{L}}(\mathbf{B})$, implying $\mathcal{L}(\mathbf{D}) \subset \hat{\mathcal{L}}(\mathbf{B})$. Now, take any $v \in \hat{\mathcal{L}}(\mathbf{B})$. By definition of the dual, we must have $v \in \text{span}(\mathbf{B})$, implying that $v = \mathbf{B}w$ for some $w \in \mathbb{R}^n$. Moreover, since the columns of \mathbf{B} are in $\mathcal{L}(\mathbf{B})$, it follows that $\mathbf{B}^\top v \in \mathbb{Z}^k$. Thus:

$$v = \mathbf{B}w = \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{B}w = \mathbf{D}(\mathbf{B}^\top v) \in \mathcal{L}(\mathbf{D}). \quad (5.4)$$

This implies $\hat{\mathcal{L}}(\mathbf{B}) \subset \mathcal{L}(\mathbf{D})$. Therefore, $\mathcal{L}(\mathbf{D}) = \hat{\mathcal{L}}(\mathbf{B})$, so \mathbf{D} is indeed the dual basis of \mathbf{B} . \square

Lemma 5.9. *If \mathbf{D} is the dual basis of \mathbf{B} , then \mathbf{B} is the dual basis of \mathbf{D} . That is, the relationship between primal and dual bases is symmetric.*

Proof. Say \mathbf{D} is the dual basis of \mathbf{B} . By Theorem 5.8, this implies $\mathbf{D} = \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1}$. Hence we have

$$\mathbf{D}^\top \mathbf{D} = (\mathbf{B}^\top \mathbf{B})^{-\top} \mathbf{B}^\top \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1} = (\mathbf{B}^\top \mathbf{B})^{-\top} = ((\mathbf{B}^\top \mathbf{B})^\top)^{-1} = (\mathbf{B}^\top \mathbf{B})^{-1}. \quad (5.5)$$

By Theorem 5.8 and (5.5), the dual basis of \mathbf{D} is:

$$\mathbf{D}(\mathbf{D}^\top \mathbf{D})^{-1} = \mathbf{D}((\mathbf{B}^\top \mathbf{B})^{-1})^{-1} = \mathbf{D}(\mathbf{B}^\top \mathbf{B}) = \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1}(\mathbf{B}^\top \mathbf{B}) = \mathbf{B}. \quad (5.6)$$

Hence, the relation between primal and dual basis is indeed symmetric. \square

Lemma 5.10. *For every lattice \mathcal{L} , we have $\det(\hat{\mathcal{L}}) = \frac{1}{\det(\mathcal{L})}$.*

Proof. Let \mathbf{B} be the basis for \mathcal{L} and \mathbf{D} be the dual basis. Then by Theorem 5.8 we have $\mathbf{B}^\top \mathbf{D} = \mathbf{B}^\top \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1} = \mathbf{I}$. By Lemma 5.9, it follows that $\mathbf{D}^\top \mathbf{B} = \mathbf{I}$. This implies that

$$(\mathbf{B}^\top \mathbf{B})(\mathbf{D}^\top \mathbf{D}) = (\mathbf{B}^\top (\mathbf{B}\mathbf{D}^\top) \mathbf{D}) = (\mathbf{B}^\top \mathbf{D}) = \mathbf{I}. \quad (5.7)$$

Hence, the Gram matrix of \mathbf{B} is the inverse of the Gram matrix of \mathbf{D} . This implies that $\det(\mathbf{B}^\top \mathbf{B}) = 1/\det(\mathbf{D}^\top \mathbf{D})$ and therefore $\det(\mathcal{L}) = 1/\det(\hat{\mathcal{L}})$. \square

We now use the results above for our main proof. Recall that the lattice we are considering is $L_{\mathbf{A}} = q\mathbb{Z}^n + \mathbf{A}\mathbb{Z}^k$. It follows from Lemmas 5.6 and 5.7 that the dual of $q\mathbb{Z}^n$ is $\frac{1}{q}\mathbb{Z}^n$. Since $q\mathbb{Z}^n \subset q\mathbb{Z}^n + \mathbf{A}\mathbb{Z}^k$, it follows that the dual of $L_{\mathbf{A}}$ must be contained in $\frac{1}{q}\mathbb{Z}^n$. This is because adding more vectors to the lattice simply adds more constraints to the dual lattice. In other words, any vector w in the dual of $L_{\mathbf{A}}$ needs to have $\langle v, w \rangle \in \mathbb{Z}$ for all $v \in q\mathbb{Z}^n \subset L_{\mathbf{A}}$, implying that $w \in \frac{1}{q}\mathbb{Z}^n$. Let $\mathbf{B} = \{b_1, \dots, b_n\}$ be a LLL-reduced basis for $L_{\mathbf{A}}$, and let $\mathbf{D} = \{d_1, \dots, d_n\}$ be the dual basis for \mathbf{B} , as defined in Theorem 5.8. Lastly, let $\mathbf{B}^* = \{b_1^*, \dots, b_n^*\}$ and $\mathbf{D}^* = \{d_1^*, \dots, d_n^*\}$ be the orthogonal bases generated by applying Gram-Schmidt to \mathbf{B} and \mathbf{D} , respectively. Since the dual lattice is contained in $\frac{1}{q}\mathbb{Z}^n$, it follows that

$$\det(\hat{\mathcal{L}}_{i+1}) = \prod_{j=i+1}^n \|d_j^*\| \geq \frac{1}{q^{n-i}}. \quad (5.8)$$

where $\hat{\mathcal{L}}_{i+1}$ is defined as the lattice spanned by d_{i+1}, \dots, d_n . Using Lemma 5.10 and Corollary 6 from [Mic07] (which explains how the relationship in Lemma 5.10 extends to the orthogonalizations of dual bases), we also know that:

$$\prod_{j=i+1}^n \|b_j^*\| = \left(\prod_{j=i+1}^n \|d_j^*\| \right)^{-1}, \quad (5.9)$$

since we can treat the last $n - i$ vectors as a partial basis. Combining (5.8) and (5.9), we get:

$$\prod_{j=i+1}^n \|b_j^*\| \leq q^{n-i} \quad (5.10)$$

$$\Rightarrow \ln \left(\prod_{j=i+1}^n \|b_j^*\| \right) \leq \ln q^{n-i} \quad (5.11)$$

$$\Rightarrow \sum_{j=i+1}^n \ln \|b_j^*\| = \sum_{j=i+1}^n \ell_j \leq (n - i) \ln q, \quad (5.12)$$

which corresponds to Fact 2.4 from [DvW21b].

5.3 Completing the Proof

Putting all of these pieces together, we have the following proposition, which corresponds to Proposition 2.5 in Ducas and van Woerden's note:

Proposition 5.11.

$$\min_i \ell_i \geq \ln(\lambda_1(\mathcal{L})) - \sqrt{2k \cdot \ln \delta' \cdot \ln q},$$

where $\delta' = \frac{1}{\sqrt{\delta}}$ for δ the traditional parameter in the LLL algorithm, as explained immediately after Definition 4.1.

Proof. Let $d = \left\lceil \sqrt{\frac{2k \cdot \ln q}{\ln \delta'}} \right\rceil$, we can then rewrite the above inequality as

$$\min_i \ell_i \geq \ln(\lambda_1(\mathcal{L})) - (d - 1) \ln(\delta'),$$

because $d - 1 \leq \sqrt{\frac{2k \cdot \ln q}{\ln \delta'}}$. Note that when $i \leq d$, the proposition follows from Equation 4.3. This is because ℓ_1 is the magnitude of a lattice vector \mathbf{b}_1^* , so $\ell_1 \geq \ln(\lambda_1(\mathcal{L}))$ by definition, and we can

iteratively apply the Lovász condition until we reach ℓ_1 :

$$\ell_i \geq \ell_{i-1} - \ln(\delta') \quad (5.13)$$

$$\geq (\ell_{i-2} - \ln(\delta')) - \ln(\delta') = \ell_{i-2} - 2 \ln(\delta') \quad (5.14)$$

$$\vdots \quad (5.15)$$

$$\geq \ell_1 - (i-1) \ln(\delta') \quad (5.16)$$

$$\geq \ln(\lambda_1(\mathcal{L})) - (i-1) \ln(\delta'). \quad (5.17)$$

It follows that for $i \leq d$, this is at least $\ln(\lambda_1(\mathcal{L})) - (d-1) \ln(\delta')$.

For the $i > d$ case, we can start by bounding the sum $\sum_{j=i}^i \ell_j$:

$$\sum_{j=1}^i \ell_j \leq \sum_{j=1}^i \ell'_j \leq i \ln(q). \quad (5.18)$$

This follows directly from the fact that $\sum_{j=1}^i \ell_j \leq \sum_{j=1}^i \ell'_j$ (Equation 4.5), and the fact that $\ell'_i \leq \ln(q)$ for all ℓ'_i (Lemma 5.1). We can also establish a lower bound by subtracting Equation 5.12 from 5.18:

$$(i-k) \ln(q) \leq \sum_{j=1}^i \ell_j \leq i \ln(q) \quad \text{for } d < i \leq n. \quad (5.19)$$

Now, we use the fact that $i > d$ and subtract

$$\sum_{j=1}^i \ell_j - \sum_{j=1}^{i-d} \ell_j \geq (i-k) \ln(q) - (i-d) \ln(q), \quad (5.20)$$

$$\sum_{j=i-d+1}^i \ell_j \geq (d-k) \ln(q). \quad (5.21)$$

Furthermore, we can iteratively apply the Lovász condition like we did above to obtain $\ell_j \leq \ell_i + (j-i) \ln(\delta')$ for $j \leq i$. Then, we can use this fact to establish an upper bound:

$$(d-k) \ln(q) \leq \sum_{j=i-d+1}^i \ell_j \leq d\ell_i + \sum_{j=0}^{d-1} j \ln(\delta'). \quad (5.22)$$

Rewriting this, we have

$$(d-k) \ln(q) \leq d\ell_i + \left(\frac{d(d-1)}{2}\right) \ln(\delta'). \quad (5.23)$$

Then, we divide everything by d and rearrange to obtain

$$\ell_i \geq \frac{(d-k) \ln(q)}{d} - \left(\frac{(d-1)}{2}\right) \ln(\delta'), \quad (5.24)$$

$$\ell_i \geq \ln(q) - \frac{k}{d} \ln(q) - \left(\frac{(d-1)}{2}\right) \ln(\delta'). \quad (5.25)$$

Recall that we defined d such that $d-1 \leq \sqrt{\frac{2k \cdot \ln q}{\ln \delta'}} \leq d$, so we can substitute and simplify:

$$\ell_i \geq \ln(q) - \frac{k}{d} \ln(q) - (d-1) \frac{1}{2} \ln(\delta'), \quad (5.26)$$

$$\ell_i \geq \ln(q) - \frac{k}{\sqrt{\frac{2k \cdot \ln q}{\ln \delta'}}} \ln(q) - \left(\sqrt{\frac{2k \cdot \ln q}{\ln \delta'}}\right) \frac{1}{2} \ln(\delta'), \quad (5.27)$$

$$\ell_i \geq \ln(q) - \frac{1}{2} \sqrt{2k \ln(q) \cdot \ln(\delta')} - \frac{1}{2} \sqrt{2k \cdot \ln(q) \cdot \ln(\delta')}. \quad (5.28)$$

Finally, we use the fact that $\lambda_1(\mathcal{L})$ must be less than q to obtain

$$\ell_i \geq \ln(\lambda_1(\mathcal{L})) - \sqrt{2k \ln(q) \cdot \ln(\delta')} \quad \text{for } d < i \leq n, \quad (5.29)$$

which is what we wanted to show. \square

All that remains is to show that Babai’s nearest plane algorithm will solve BDD. Recall that Babai’s algorithm solves γ -CVP for $\gamma(n) \leq 2^{n/2}$ [Reg04b], and when $r < \lambda_1(\mathcal{L})/2$, then CVP solves BDD because the solution is unique. In other words, there is only one possible vector within r in this case, and that is the closest one. Moreover, the smallest Gram-Schmidt vector $\min_i \|b_i^*\|$ is less than $\lambda_1(\mathcal{L})$ [Bai06], and therefore Babai’s algorithm will solve BDD for $r < \min_i \|b_i^*\|$. Due to the exponential factor and the above proposition, we have shown that Babai’s algorithm can solve BDD up to radius $\frac{1}{2}\lambda_1(\mathcal{L}) \cdot \exp\left\{-\sqrt{2k \cdot \ln(q) \cdot \ln(\delta')}\right\}$, concluding the proof of Theorem 1.3.

6 Conclusion and Future Work

After a survey of numerous interrelated lattice problems and previous work on quantum approaches to these problems, we have presented an overview of Eldar and Hallgren’s work, as well as a complete proof of Ducas and van Woerden’s result with all details filled in.

There is further work to be done in the area of classical algorithms for BDD and related lattice problems. In particular, instead of LLL, it would be worthwhile to consider alternative basis reductions, such as a Block Korkin-Zolotarev (BKZ) reduction, which is known to give a better approximation factor than LLL when solving γ -CVP using Babai’s nearest plane algorithm [NV10]. It would also be interesting to consider randomized heuristics for lattice problems, such as Klein’s algorithm [LLS11].

There is a more general question of when a quantum algorithm for a lattice problem is guaranteed to have a classical counterpart. If more quantum-classical pairings are found, it would be interesting to consider the connections between these cases and conjecture such a result.

Acknowledgements

We would like to thank Professor Boaz Barak and Emil Khabibouline for their guidance throughout this project. We are also thankful to Professor Vinod Vaikuntanathan and Jessica Sorrell for useful correspondence on lattice-based quantum algorithms and their classical counterparts, and to Professor Sean Hallgren and Dr. Lior Eldar for helpful feedback and for kindly providing us with the manuscript of their result.

References

- [Ajt98] Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions. In *Proceedings of the 13th annual ACM Symposium on Theory of Computing*, pages 10–19, 1998.
- [Bab86] László Babai. On Lovász’ lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
- [Bai06] Thomas Baigneres. Cryptosystems and LLL. Technical report, 2006.
- [Bar22] Boaz Barak. *An Intensive Introduction to Cryptography*. 2022.
- [DD18] Daniel Dadush and Léo Ducas. Lecture 4 for the course “Intro to Lattice Algorithms and Cryptography”. <https://homepages.cwi.nl/~dadush/teaching/lattices-2018/notes/lecture-4.pdf>, 2018.
- [Den16] Xinyue Deng. An introduction to Lenstra-Lenstra-Lovasz Lattice basis reduction algorithm, 2016.
- [DvW21a] Léo Ducas and Wessel van Woerden. A note on a Claim of Eldar & Hallgren: LLL already solves it (Version 1). *Cryptology ePrint Archive*, 2021.

- [DvW21b] Léo Ducas and Wessel van Woerden. A note on a Claim of Eldar & Hallgren: LLL already solves it (Version 2). *Cryptology ePrint Archive*, 2021.
- [EH22] Lior Eldar and Sean Hallgren. An efficient quantum algorithm for lattice problems achieving subexponential approximation factor. *arXiv preprint arXiv:2201.13450*, 2022.
- [Gal12] Steven D Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012.
- [GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and J-P Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.
- [Hal21] Sean Hallgren. An efficient quantum algorithm for lattice problems achieving subexponential approximation factor. Available at https://www.youtube.com/watch?v=K5Apl_qCnDA, 2021.
- [Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.
- [Kup11] Greg Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *arXiv preprint arXiv:1112.3333*, 2011.
- [LLL82] Arjen K. Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(ARTICLE):515–534, 1982.
- [LLS11] Shuiyin Liu, Cong Ling, and Damien Stehlé. Decoding by sampling: A randomized lattice algorithm for bounded distance decoding. *IEEE Transactions on Information Theory*, 57(9):5933–5945, 2011.
- [LM09] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *Annual International Cryptology Conference*, pages 577–594. Springer, 2009.
- [MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671. Springer Science & Business Media, 2002.
- [Mic01] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, 2001.
- [Mic07] Daniele Micciancio. Lecture 3 for the course “Lattice Algorithms and Applications”. <https://cseweb.ucsd.edu/classes/sp07/cse206a/lec3.pdf>, 2007.
- [NC10] Michael A Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [NV10] Phong Q. Nguyen and Brigitte Vallée. *The LLL algorithm*. Springer, 2010.
- [Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 333–342, 2009.
- [Pei13] Chris Peikert. Lecture 3 for the course “Lattices in Cryptography”. <https://web.eecs.umich.edu/~cpeikert/lic13/lec03.pdf>, 2013.
- [Reg04a] Oded Regev. Lecture 2 for the course “Lattices in Computer Science”. https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/lll.pdf, 2004.
- [Reg04b] Oded Regev. Lecture 3 for the course “Lattices in Computer Science”. https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/cvp.pdf, 2004.
- [Reg04c] Oded Regev. Quantum computation and lattice problems. *SIAM Journal on Computing*, 33(3):738–760, 2004.
- [Reg04d] Oded Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. *arXiv preprint quant-ph/0406151*, 2004.
- [Reg10] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 7(30):11, 2010.
- [SD16] Noah Stephens-Davidowitz. Lecture 5 for the course “Lattices Mini Course”. http://www.noahsd.com/mini_lattices/05__babai.pdf, 2016.

- [SD21] Noah Stephens-Davidowitz. Dimension-preserving reductions between lattice problems, 2015. 2021.
- [vEB81] Peter van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. *Technical Report, Department of Mathematics, University of Amsterdam*, 1981.