Random walks on Arakelov class groups

Boer, K. de

Citation

Boer, K. de. (2022, September 22). *Random walks on Arakelov class groups*. Retrieved from https://hdl.handle.net/1887/3463719

3. The Continuous Hidden Subgroup Problem

3.1. Summary

This chapter is about a complexity analysis of a slightly modified algorithm of Eisenträger et al. [Eis+14] that quantumly solves the *continuous hidden subgroup problem*. This problem consists of finding a 'hidden lattice' Λ in $\mathbb{R}^m$ given a (possibly) quantum function $\mathbf{f} : \mathbb{R}^m \to \mathcal{S}$ that is periodic with respect to this lattice Λ. This computational problem falls into the class of the so-called 'period finding problems'.

This quantum algorithm mimics the blueprint of Shor's algorithm for finding a hidden subgroup H in a discrete abelian group G, given an oracle function on the group that is strictly periodic with respect to H. This blueprint consists of consecutively sampling a uniform quantum superposition over all group elements, applying an oracle call to the H-periodic function, and computing a discrete quantum Fourier transform. Then, one measures to obtain a character $\chi \in \hat{G}$ that has H in its kernel. Assembling enough of such characters allows to retrieve H itself.

The quantum algorithm solving the continuous hidden subgroup problem

The quantum algorithm of this chapter deviates from this blueprint in a few ways. (1) Since the ambient group $\mathbb{R}^m$ is continuous, we need to cut-off and discretize this space to get something finite and thus processable by a quantum computer. This has as a consequence that the Fourier transform becomes discretized as well, inducing errors with respect to the continuous Fourier transform. (2) The initial state of this quantum algorithm does not consist of a uniform quantum sample but of a Gaussian state instead. This is done to ease the analysis, as both the Gaussian function and its Fourier transform (which is also a Gaussian function) have tight tail bounds. (3) The measurement output is, due to the cut-off and discretization, always an *approximation* of a dual lattice vector $\ell^* \in \Lambda^*$ (which can be seen as a character of $\mathbb{R}^m$ with Λ in its kernel). So, in the end, we cannot expect more to gain from this algorithm than an *approximate basis* $\tilde{B}$ of the lattice Λ. (4) Such an approximate basis is obtained as follows. By sampling many approximate $\ell^* \in \Lambda^*$, LLL-reducing these samples to an approximate basis $\tilde{D}$ of the dual lattice Λ^*, and inverting and transposing $\tilde{D}$, one retrieves an approximate basis $\tilde{B}$ of Λ.

Analysis of the algorithm

Each deviation from the original 'hidden subgroup problem blueprint' causes difficulties; mostly those difficulties take the shape of *discretization errors*. We show how to solve these difficulties per deviation. Tackling these difficulties was already partially done by Eisenträger et al. [Eis+14]; we revisit their work to obtain a more explicit and precise complexity.

(1) The discrete Fourier transform and the continuous (real) Fourier transform can be shown to differ not too much if their input function is *continuous enough*. A large part of this chapter (Section 3.5) is devoted to show that if the Λ-periodic oracle function is *Lipschitz continuous*, the induced error by using a discrete Fourier transform instead of a continuous one can be reasonably bounded.

(2) For the initial input to be Gaussian, one needs to know how to actually assemble this state on a quantum computer. Such a Gaussian superposition has already been shown to be computable in polynomial time by Kitaev and Webb [KW08], but for completeness we included a more precise complexity estimate in Appendix A.5.

(3) Due to the discrete nature of the quantum algorithm, the output dual lattice point can only be approximated within a certain distance. The maximum allowed distance (relative to the minimum distance $\lambda_1(\Lambda^*)$ of the dual lattice Λ^*) will be a parameter in the algorithm, called $\delta > 0$.

One of the problems that might occur is that the output dual lattice points are not equidistributed enough on Λ^*, thus not giving enough information to retrieve a basis of Λ. An extra assumption on the Λ-periodic function $\mathbf{f}$ is needed to avoid such a situation; which we call *separating*. A separating Λ-periodic function can be intuitively thought of as being not too constant. Showing that such an oracle will yield equidistributed points in Λ^* is the object of Section 3.6.

(4) From many such δ-close dual lattice points one can compute an approximate basis of the dual lattice Λ^* by means of LLL-reduction; from this approximate dual basis one can obtain a basis of Λ by inversion and transposition. These operations (LLL-reduction and inversion) are quite *numerical unstable*, meaning that they make existing errors in the input progressively larger. Using a result of Buchmann and Kessler [BK96] one can reasonably bound the final error (see Section 3.7).

Relation with the Arakelov (ray) class group

The computation of the Arakelov (ray) class group can be phrased in terms of a hidden lattice problem; a fact that can already be inferred from the original applications of the hidden lattice problem, namely computing (S)-unit groups and class groups [BS16; Eis+14] in works of Biasse, Song and Eisenträger et al. By a slight modification in formulation of the ideas in these papers one can construct an oracle on the Arakelov divisor group that is periodic with respect to the principal divisors. In this modification, a ‘reduced’ version of the Arakelov (ray) divisor group is used, one with only finitely many prime ideals, that are required to generate the ideal class group. Finding the periodicity of this oracle then allows to find explicit relations that define the Arakelov (ray) class group.

At the time of writing, a precise complexity estimation (beyond polynomial time) of the oracle function in this approach to quantumly compute Arakelov (ray) class groups is still open.

3.2. Introduction

The Hidden Subgroup Problem

Among all quantum algorithms, Shor's algorithm [Sho94] for factoring and finding discrete logarithms is singular by its cryptanalytic implications. Due to progress toward the realization of large quantum computers, this celebrated algorithm is now motivating the standardization of quantum-resistant schemes [Nat17], in preparation of a global update of widely deployed encryption and authentication protocols.

The core idea of quantum period finding [Sho94] is not limited to factoring and discrete logarithm. The *Hidden Subgroup Problem*, formalized in [ME98], serves as a convenient interface between the quantum-algorithmic techniques for period finding, and applications to solve other computational problems, in particular problems arising from number theory. We will here discuss only the case of commutative groups. The cases of non-abelian groups such as dihedral groups are very interesting as well and have fascinating connections with lattice problems [Reg04b]; however, no polynomial time algorithm is known for those cases, and the best known algorithm has sub-exponential complexity [Kup05], using very different techniques.

The simplest version of the Hidden Subgroup Problem consists of finding a hidden subgroup H in a *finite* abelian group G, when given access to a strictly H-periodic function $\mathbf{f} : G \to S$. Here, in the language of representation theory, the off-the-shelf period-finding quantum algorithm finds a uniformly random character $\chi \in \hat{G}$ that acts trivially on H. Shor's original algorithm [Sho94] for integer factoring finds a hidden subgroup H in the ambient group $\mathbb{Z}$. The infiniteness of $\mathbb{Z}$ induces some "cut-off" error;

nevertheless, the distribution of the algorithm's output is still concentrated around the multiples of the inverse period.

A generalization to the real line $H = \mathbb{R}$ was given by Hallgren [Hal07] and allows to solve Pell's equation. The case of real vector space of constant dimension $H = \mathbb{R}^c$ has also been studied [Hal05; SV05], and permits the computation of unit groups of number fields of fixed finite degree.

The *Continuous* Hidden Subgroup Problem

The latest generalization of the HSP algorithm, given by Eisenträger, Hallgren, Kitaev and Song in an extended abstract [Eis+14], targets the ambient group $G = \mathbb{R}^m$ (for a non-constant dimension m) with a hidden discrete subgroup $H = \Lambda$, i.e. a *lattice*. Next to the ambient group $\mathbb{R}^m$ being *continuous*, an additional special feature is that the Λ-periodic function $\mathbf{f}$ is assumed to produce a "quantum output". More formally, $\mathbf{f} : \mathbb{R}^m \to \mathcal{S}$, $x \mapsto |\mathbf{f}(x)\rangle$, where $\mathcal{S}$ is the state space of a quantum system, and the HSP algorithm is given access to a unitary that maps $|x\rangle|0\rangle$ to $|x\rangle|\mathbf{f}(x)\rangle$. A crucial observation here is that $|\mathbf{f}(x)\rangle$ and $|\mathbf{f}(y)\rangle$ are *not* necessarily orthogonal (or even distinct) for distinct x and y modulo Λ. In other words, it is not assumed that $\mathbf{f}$ is *strictly* periodic, but merely that $|\mathbf{f}(x)\rangle$ and $|\mathbf{f}(y)\rangle$ are "somewhat orthogonal" for x and y that are "not too close" modulo Λ, and that $\mathbf{f}$ is Lipschitz continuous.

More specifically, they consider a variation of the standard HSP algorithm in order to tackle the Continuous Hidden Subgroup Problem (CHSP). In order to deal with the continuous nature of the domain $\mathbb{R}^m$ of $\mathbf{f}$, the given HSP algorithm acts on a bounded "grid" of points within $\mathbb{R}^m$. Additionally, the algorithm is modified in the following ways: (1) The initial state is not a uniform superposition (over the considered grid points in $\mathbb{R}^n$) but follows a trigonometric distribution, and (2) the quantum Fourier transform is done "remotely", i.e., rather than applying it to the actual register, the register is entangled with an ancilla and the quantum Fourier transform is then applied to the ancilla instead. According to Eisenträger et al. [Eis+14], applying the quantum Fourier transform directly would make the resulting approximation errors difficult to analyze.

As an application, Eisenträger et al. also gave a quantum polynomial time algorithm for computing the unit group of a number field in their article [Eis+14]. This was generalized by Biasse and Song [BS16] to the computation of S-unit groups, and therefore to the computation of class groups and to finding a generator of a principal ideals. This led to solving the shortest vector problem in certain ideal lattices for non-trivial approximation factors [Cra+16; CDW17; PHS19]. While the cryptanalytic consequences for ideal-lattice based cryptography seem limited so far [DPW19], these results demonstrate a hardness gap between ideal lattices and general ones.

Our Contributions

The goal of this chapter is to provide a complete, modular, and quantitative analysis of (a slightly modified version of) the Continuous HSP quantum algorithm given by [Eis+14]. More concretely, we provide an explicit bound on the number of qubits needed by the algorithm, clarifying the dependency on the parameters of the Continuous HSP instance and on the required precision and success probability. This shows explicitly in what parameters the algorithm is polynomial time and with what exponent.

The algorithm that we consider and analyze differs from the one considered in [Eis+14] in the following points:

- First, we specify the initial state of the algorithm to have Gaussian amplitudes, while [Eis+14, Sec. 6.2] suggests to use a cropped trigonometric function; as far as we can see, our choice makes the analysis simpler and tighter thanks to the well known tail-cut and smoothness bounds of Banaszczyk [Ban93] and Micciancio and Regev [MR07].
- Secondly, we do not make use of a “remote” Fourier transform but instead follow the blueprint of Shor’s original algorithm in that respect; the claimed advantage of the “remote” Fourier transform is unclear to us.

These modifications simplify the algorithm and its analysis. Due to the lack of details given in [Eis+14], we can not state a complexity comparison, but we think this variation is at least as efficient as the original algorithm.

Our analysis is divided into four parts, each summarized by a formal statement given in Sections 3.3.3 to 3.3.6, leading to the main theorem (Section 3.3.2). We insist on this modular presentation, so as to enable future work on optimization and specialization of this algorithm to instances of interests; specific suggestions follow.

Dual lattice sampling. In the first part, which is the technically more involved one, we show that the appropriately discretized and finitized, but otherwise (almost) standard HSP quantum algorithm produces sample points in $\mathbb{R}^m$ that lie close to the dual lattice Λ^* with high probability. More precisely, and more technically speaking, we show that the algorithm's output is a sample point close to $\ell^* \in \Lambda^*$ with probability close to $\langle c_{\ell^*} | c_{\ell^*} \rangle$, where the vectors $|c_{\ell^*}\rangle$ are the Fourier coefficients of the function $\mathbf{f}$. This is in line with the general HSP approach, where for instance Shor's algorithm for period finding over $\mathbb{Z}$ produces a point that is close to a random multiple of the inverse period, except with bounded probability.

In this first part (Section 3.4 and Section 3.5), we bound the complexity of the core algorithm in terms of the error probability that we allow in the above context of a sampling algorithm, and depending on the Lipschitz constant of $\mathbf{f}$. In particular, we show that the number of qubits grows as mQ, where Q, the "number of qubits per dimension", grows linearly in the logarithm of the Lipschitz constant of $\mathbf{f}$, the logarithm of the inverse of the error probability and the logarithm of the inverse of the (absolute) precision, and quasi-linearly in m. The running time of the algorithm is then bounded by $O(mQ\log(mQ))$, by using an approximate Fourier transform [HH00].

Full dual recovery. In the second part, Section 3.6, we then relate the parameters of the Continuous HSP instance to the number of sample points, and thus to how often the core algorithm needs to be repeated, necessary in order to have an approximation of the entire dual lattice Λ^*.

Primal basis reconstruction. In the third part, Section 3.7, we study the numerical stability of reconstructing an approximate basis of the primal lattice Λ from a set of approximate generators of the dual lattice Λ^*. This is based on the Buchmann-Pohst algorithm [BK96] already mentioned in [Eis+14]. The claim of [Eis+14] involves intricate quantities related to sublattices of Λ, making the final complexity hard to derive; we provide a simpler statement with a detailed proof.

Gaussian state preparation. Finally, in Appendix A.5, we revisit the quantum polynomial-time algorithm for the preparation of the Gaussian initial state [GR02; KW08] used as a black-box in our first part, and provide its precise complexity.

Conclusion. These four parts lead to our formal and quantitative version of the informal CHSP Theorem of Eisenträger et al. [Eis+14, Thm. 6.1], stated as Theorem 3.3 in Section 3.3.2.

Conclusion and Research Directions

Our conclusion is that, in its generic form, the Continuous Hidden Subgroup Problem is rather expensive to solve; not accounting for other parameters than the dimension m, it already requires $\tilde{O}(m^3)$ qubits and $\tilde{O}(m^4)$ quantum gates (using an approximate quantum Fourier transform). However, this inefficiency seems to be a consequence of its genericity. In particular, the core algorithm for Dual Lattice Sampling would only need $\tilde{O}(m^2)$ qubits, if it wasn't for accommodating for the terrible numerical stability of the Primal Basis Reconstruction step. Similarly, we expect the number of samples needed to generate the dual lattice to be significantly smaller for smoother oracle functions.

All in all, our modular analysis of the generic steps of the CHSP algorithm sets the stage for analyzing and optimizing its specializations, in particular to cryptanalytic applications [Cra+16; CDW17]. We propose as few research directions towards this objective:

- Study the costs (qubits, quantum gates) and the parameters of the oracle functions from [Eis+14; BS16; Son13] for solving the Unit Group Problem, the Principal Ideal Problem (PIP), and for the computation of the class group.
- Find stronger hypotheses satisfied by the above oracle functions (or by variant thereof) that improve this generic analysis of the CHSP algorithm; or resort to an ad-hoc analysis of the Full Dual Recovery step by directly studying the spectrum of these oracle functions.
- Explore the possibility of a trade-off between the (classical) Primal Basis Reconstruction step and the (quantum) Dual Lattice Sampling step, possibly up to small sub-exponential classical complexity. More specifically, does replacing LLL by BKZ with a medium block-size substantially improve the numerical stability of Buchmann-Pohst algorithm?
- Exploit prior knowledge of sublattices (potentially close to full-rank) of the hidden lattice to accelerate or skip the Full Dual Recovery and Primal Basis Reconstruction steps. This is for example the case when solving PIP [BS16] while already knowing the unit group and the class group of a given number field. This would be applicable in the context of [Cra+16; CDW17].
- Exploit known symmetries of the hidden sublattice to improve the Full Dual Recovery and Primal Basis Reconstruction steps. Such symmetries are for example induced by the Galois action on the log-unit lattice and the lattice of class relation, in particular in the case of the cyclotomic number fields. This would again be applicable in the context of [Cra+16; CDW17].

Remark 3.1. *Recovering the* exact *hidden lattice is outside the scope of this work, since this task is application-dependent. It is even true that one* *cannot generally expect the quantum algorithm of this chapter to recover the exact hidden lattice, without extra information about this hidden lattice.*

For instance, when applying this algorithm to compute the unit group $\mathcal{O}_K^\times$ *of a number field* K*, the hidden lattice will be the so-called logarithmic unit lattice. Of this lattice it is known that any point is of the shape* $\operatorname{Log}(\eta) = (\log|\sigma(\eta)|)_\sigma \in \operatorname{Log} K_\mathbb{R}^0$ *with* $\eta \in \mathcal{O}_K^\times \subseteq \mathcal{O}_K$*; its entries are logarithms of integral elements in a given number field. This is the extra information that is to be exploited in order to get the exact lattice. Namely, from a sufficiently good approximation of the logarithm of a unit one can obtain the exact underlying unit, simply by taking the exponential and rounding it to the closest element in the ring of integers* $\mathcal{O}_K$*.*

3.3. Problem Statements and Results

3.3.1. Notation and Set-up

Here and throughout this chapter, $\mathcal{H}$ is a complex Hilbert space of dimension $N = 2^n$, and $\mathcal{S}$ is the unit sphere in $\mathcal{H}$; thus, a vector in $\mathcal{S}$ describes the state of a system of n qubits. For an arbitrary positive integer m, we consider a function

$$\mathbf{f} : \mathbb{R}^m \to \mathcal{S} \subset \mathcal{H}\,,\ x \mapsto |\mathbf{f}(x)\rangle$$

that is periodic with respect to a full rank lattice $\Lambda \subset \mathbb{R}^m$; hence, $\mathbf{f}$ may be understood as a function $\mathbb{R}^m/\Lambda \to \mathcal{S}$. The function $\mathbf{f}$ is assumed to be Lipschitz continuous with Lipschitz constant

$$\operatorname{Lip}(\mathbf{f}) = \inf\Big\{L > 0 \mid \big\||\mathbf{f}(x)\rangle - |\mathbf{f}(y)\rangle\big\|_{\mathcal{H}} \leq L\|x-y\|_{2,\mathbb{T}^m}\Big\}.$$

Later, we will also require $\mathbf{f}$ to be “sufficiently non-constant”. One should think of $\mathbf{f}$ as an oracle that maps a classical input x to a quantum state over n qubits, which is denoted $|\mathbf{f}(x)\rangle$.

We write Λ^* for the dual lattice of Λ. By $\lambda_1(\Lambda)$ we denote the length of a shortest non-zero vector of Λ, and correspondingly for $\lambda_1(\Lambda^*)$. Since Λ is typically clear from the context, we may just write λ_1 and λ_1^* instead of $\lambda_1(\Lambda)$ and $\lambda_1(\Lambda^*)$.

We denote by $\mathcal{B}_r(x) = \{y \in \mathbb{R}^m \mid \|y - x\| < r\}$ the open Euclidean ball with radius r around x. For the open ball around 0 we just denote $\mathcal{B}_r$, and for a set $X \subset \mathbb{R}^m$ we write $\mathcal{B}_r(X) = \bigcup_{x \in X} \mathcal{B}_r(x)$.

Definition 3.2 (Definition 1.1 from [Eis+14]). *A function* $\mathbf{f} : \mathbb{R}^m \to \mathcal{S} \subset \mathcal{H}$ *is said to be an* (a, r, ϵ)*-HSP oracle of the full-rank lattice* $\Lambda \subset \mathbb{R}^m$ *if*

- $\mathbf{f}$ *is* Λ*-periodic,*
- $\mathbf{f}$ *is* a*-Lipschitz:* $\mathrm{Lip}(f) \leq a$*,*
- $\mathbf{f}$ *is* (r, ϵ)-separating *(see Figure 3.1): I.e.,* $|\langle \mathbf{f}(x) | \mathbf{f}(y) \rangle| \leq \epsilon$ *for all* $x, y \in \mathbb{R}^m$ *satisfying* $d_{\mathbb{R}^m/\Lambda}(x, y) \geq r$*.*

where $d_{\mathbb{R}^m/\Lambda}(x, y) = \min_{v \in \Lambda} \|x - y - v\|$ *denotes the distance induced by the Euclidean distance of* $\mathbb{R}^n$ *modulo* Λ*.*

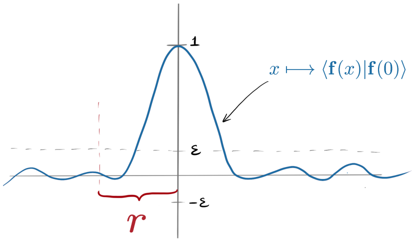

Figure 3.1.: A picture of what an (r, ϵ)-separating function $\mathbf{f}$ should look like: outside of the interval or length $2r$ around the origin, the inner product $x \mapsto \langle \mathbf{f}(x) | \mathbf{f}(0) \rangle$ deviates from 0 by no more than ϵ.

3.3.2. Main Theorem: Continuous Hidden Subgroup Problem

Theorem 3.3. *There exists a quantum algorithm that, given access to an* (a, r, ϵ)*-HSP oracle with period lattice* Λ*,* $r < \lambda_1(\Lambda)/6$ *and* $\epsilon < 1/4$*, computes, with constant success probability, an approximate basis* $\tilde{B} = B + \Delta_B$ *of this lattice* Λ*, satisfying* $\|\Delta_B\| < \tau$*.*

This algorithm makes k quantum oracle calls to the (a, r, ϵ)-HSP oracle, and uses $mQ + n$ qubits, $O(kmQ \cdot (\log(kmQ))^2)$ quantum gates and poly$(m, \log \frac{a}{\lambda_1^*}, \log \frac{a}{\tau})$ classical bit operations, where

$$Q = O(mk) + O\left(\log \frac{a}{\lambda_1^*}\right) + O\left(\log \frac{1}{\lambda_1^* \cdot \tau}\right), \tag{3.24}$$

$$k = O\left(m \cdot \log\left(\sqrt{m} \cdot a \cdot (\det \Lambda)^{1/m}\right)\right) \tag{3.25}$$

Remark 3.4. *Note that the quantities inside logarithms are homogeneous. In particular, scaling the lattice Λ by a factor f, also scales τ, $1/a$ and $1/\lambda_1^*$ by the same factor f, leaving the complexity parameters Q and k unaffected.*

Remark 3.5. *The expert reader may expect the "distortion" parameter $\lambda_1 \cdot \lambda_1^*$ of the lattice Λ to have a bearing on the complexity of this algorithm. It is indeed implicitly the case: the assumption the HSP definition implies that $ar \geq \mathrm{Lip}(\mathbf{f}) \cdot r \geq 1 - \epsilon$ (see Figure 3.2), and therefore the theorem's hypothesis requires $a \geq \mathrm{Lip}(\mathbf{f}) \geq \frac{9}{4\lambda_1}$.*

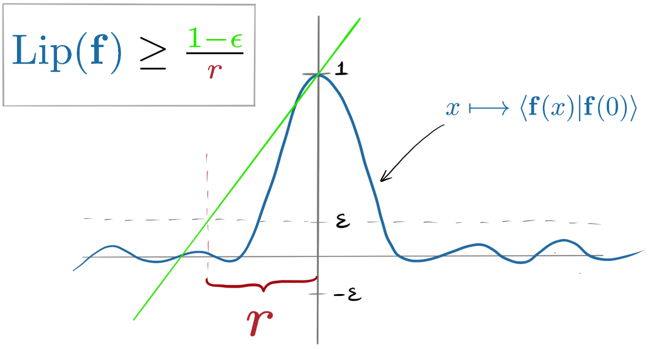
$\mathrm{Lip}(\mathbf{f}) \geq \frac{1-\epsilon}{r}$
1
$x \longmapsto \langle \mathbf{f}(x) | \mathbf{f}(0) \rangle$
ε
-ε
r

Figure 3.2.: Due to the (r, ϵ)-separating property of the oracle function $\mathbf{f}$, its Lipschitz constant cannot be too small.

The proof of Theorem 3.3 can be found in Section 3.3.7.

3.3.3. Dual Lattice Sampling Problem

Following our modular approach as outlined in the introduction, we first consider the following *Dual Lattice Sampling Problem*. Informally, the task is to sample points in $\mathbb{R}^m$ that are respectively close to points $\ell^* \in \Lambda^*$ that follow the distribution $\mathcal{D}_{ideal}(\ell^*) = \langle c_{\ell^*} | c_{\ell^*} \rangle$, where $|c_{\ell^*}\rangle$ are the vectorial Fourier coefficients of $\mathbf{f} : \mathbb{R}^m/\Lambda \to \mathcal{S}$ (see Section 2.2.4).

Problem 3.6 (Dual Lattice Sampling Problem). *Given error parameter $\eta > 0$ and a relative distance parameter $\frac{1}{2} > \delta > 0$, and given oracle access to an HSP oracle $\mathbf{f}$ as above, sample according to a (finite) distribution $\mathcal{D}$ on $\mathbb{R}^m$ that satisfies, for any $S \subseteq \Lambda^*$,*

$$p_S := \mathcal{D}(\mathcal{B}_{\delta\lambda_1^*}(S)) \geq \left(\sum_{\ell^* \in S} \langle c_{\ell^*} | c_{\ell^*} \rangle \right) - \eta \,. \tag{3.26}$$

In the problem statement above, $\mathcal{D}(\mathcal{B}_{\delta\lambda_1^*}(S))$ denotes the cumulative weight of the set $\mathcal{B}_{\delta\lambda_1^*}(S) = \bigcup_{s \in S} \mathcal{B}_{\delta\lambda_1^*}(s)$ with respect to the distribution $\mathcal{D}$. Here, $\mathcal{B}_{\delta\lambda_1^*}(s) = \{y \in \mathbb{R}^m \mid \|s - y\| < \delta\lambda_1^*\}$ is the open ball of radius $\delta\lambda_1^*$ around $s \in S \subseteq \Lambda^* \subseteq \mathbb{R}^m$.

Theorem 3.7. *Algorithm 2 solves the Dual Lattice Sampling Problem with parameters η and δ; it uses one call to the Gaussian superposition subroutine (see Theorem 3.12), one quantum oracle call to $\mathbf{f}$, $mQ + n$ qubits, and $O(mQ\log(mQ))$ quantum gates, where*

$$Q = O\left(m \log\left(m\right)\right) + O\left(\log\left(\frac{a}{\eta \cdot \delta\lambda_1^*}\right)\right). \tag{3.27}$$

Remark 3.8. *Note that this step only requires smoothness of the HSP oracle (via the Lipschitz constant), but does not rely on the "separateness" assumption (third item of Definition 3.2). Indeed this third assumption will only play a role to ensure that those samples are actually non-trivial and usable.*

3.3.4. Full Dual Lattice Recovery

Recovering the full lattice (or, equivalently, its dual) requires an extra assumption on the oracle function $\mathbf{f}$, as captured by the third condition in the following definition, reformatted from Definition 1.1 of [Eis+14].

According to Eisenträger et al. [Eis+14], for (some undetermined) adequate parameters, Definition 3.2 ensures that the distribution on the dual lattice Λ^* is not concentrated on any proper sublattice, hence sufficiently many samples will generate the lattice fully. We formalize and quantify this proof strategy, and obtain the following quantitative conclusion. We note that the constraints on r and ϵ are milder than one could think, for example ϵ does not need to tend to 0 as a function of n or m. More precisely, a constant $\epsilon < 1/4$ and a constant $r \leq \lambda_1(\Lambda)/6$ would suffice.

Theorem 3.9. *Let* $\mathbf{f} : \mathbb{R}^m \to \mathcal{S}$ *be an* (a, r, ϵ)*-HSP oracle of the full-rank lattice* $\Lambda \subset \mathbb{R}^m$*, with* $r \leq \lambda_1(\Lambda)/6$ *and* $\epsilon < 1/4$*. Let* $\mathcal{D}_{\mathbf{f}}$ *be the distribution supported by* Λ^**, with weight* $\langle c_{\ell^*} | c_{\ell^*} \rangle$ *at* $\ell^* \in \Lambda^*$*, where* $|c_{\ell^*}\rangle$ *are the vectorial Fourier coefficients of the function* $\mathbf{f}$*.*
Then, with overwhelming probability, we need at most

$$O\Big(m \log_2\left(ma \cdot \det(\Lambda)^{1/m}\right)\Big)$$

samples from $\mathcal{D}_{\mathbf{f}}$ *to fully generate the lattice* Λ^**.*

The above theorem is obtained by combining Lemma 3.21 and proposition 3.24 from Section 3.6, instantiating the parameter R to $R^2 = ma^2$. This choice is somewhat arbitrary and given for concreteness, however it does not have a critical quantitative impact.

3.3.5. Primal Basis Reconstruction

Theorem 3.10. *There exists a polynomial time algorithm, that, for any matrix* $G \in \mathbb{R}^{k \times m}$ *of* k *generators of a (dual) lattice* Λ^**, and given an* *approximation* $\tilde{G} = G + \Delta_G \in \mathbb{Q}^{k \times n}$*, computes an approximation* $\tilde{B} = B + \Delta_B$ *of a basis* B *of the primal lattice* Λ*, such that*

$$\|\Delta_B\|_\infty \leq \frac{2^{O(mk)} \cdot \|\tilde{G}\|_\infty^{m+1}}{(\lambda_1^*)^3 \cdot \det(\Lambda^*)} \cdot \|\Delta_G\|_\infty,$$

under the assumption that $\|\Delta_G\|_\infty < \frac{\min(1,(\lambda_1^*)^2) \cdot \det(\Lambda^*)}{2^{O(km)} \cdot \|\tilde{G}\|_\infty^{m+1}}$.

Remark 3.11. *More specifically, the algorithm from Theorem 3.10 essentially consists of the Buchmann-Pohst algorithm [BP89; BK96] and a matrix inversion. Its complexity is dominated by two calls to LLL on matrices of dimension* $(m+k) \times k$ *and entry bit size* $O(k^2 \log(\|\tilde{G}\|/\lambda_1^*))$ *(see the discussion before [BK96, Cor. 4.1]). One can optimize the final running time by choosing a fast variant of LLL, e.g., [NS16].*

Our contribution on this step is merely a completed numerical analysis, with the help of a theorem from [CSV12]. A claim with a similar purpose is given in [Eis+14], yet involves more intricate lattice quantities.

3.3.6. Gaussian State Preparation

The main algorithm of this paper requires the preparation of a multidimensional Gaussian initial state, which can be obtained by generating the one-dimensional Gaussian state on m parallel quantum registers. This task is known to be polynomial time [GR02; KW08], and we provide a quantitative analysis in Appendix A.5. The precise running time of preparing this Gaussian state is summarized below.

Theorem 3.12. *For* $q = 2^Q \in \mathbb{N}$*, error parameter* $\eta \in (0,1)$ *and* $s > 2\sqrt{\log(m/\eta)}$*, there exists an quantum algorithm that prepares the higher-dimensional Gaussian state*

$$\frac{1}{\sqrt{\rho_{1/s}(\mathbb{D}_{\mathrm{rep}}^m)}} \sum_{\mathbf{x} \in \mathbb{D}_{\mathrm{rep}}^m} \sqrt{\rho_{1/s}(\mathbf{x})} |\mathbf{x}\rangle = \bigotimes_{j=1}^{m} \frac{1}{\sqrt{\rho_{1/s}(\frac{1}{q}[q]_c)}} \sum_{x \in \frac{1}{q}[q]_c} \sqrt{\rho_{1/s}(x)} |x\rangle,$$

within trace distance η*, using* $O(mQ + \log(\eta^{-1}))$ *qubits and using* $O(mQ \cdot \log(mQ\eta^{-1})^2)$ *quantum gates.*

Remark 3.13. *In Theorem 3.3, we chose η to be $1/k^2$. Therefore, one call to the m-dimensional Gaussian state preparation with the parameters of Theorem 3.3 takes $O(mQ+\log(k))$ qubits and $O(mQ\log(kmQ)^2)$ quantum gates. As Theorem 3.3 requires k subsequent preparations of the m-dimensional Gaussian state, the total costs of the Gaussian state preparation steps are $O(mQ+\log(k))$ qubits (by reusing qubits) and $O(kmQ\log(kmQ)^2)$ quantum gates.*

This is slightly more than the costs of k times applying the Fourier transform, and it explains the quantum gate complexity of $O(kmQ\log(kmQ)^2)$ in Theorem 3.3.

3.3.7. Proof of the Main Theorem

Proof of Theorem 3.3. The result is obtained by running Algorithm 1 and instantiating Theorems 3.7, 3.9, 3.10 and 3.12.

Correctness of Algorithm 1. In step one, the dual sampling algorithm (Algorithm 2) is applied k times with error probability $\eta = 1/k^2$. The probability that all measurements are actually $\delta\lambda_1^*$-close to dual lattice points and are of length less than $\sqrt{m}a$ is then at least $(1-\eta)^k = (1-1/k^2)^k \geq 1-1/k$, which is at least a constant success probability. We assume in the rest of the proof that all measurements are indeed $\delta\lambda_1^*$-close to dual lattice points and of length less than $\sqrt{m}\cdot a$.

In step two, these $\delta\lambda_1^*$-close-to-Λ^* samples are assembled into a matrix $k\times m$-matrix $\tilde{G}$, on which is then applied the Buchmann-Pohst algorithm [BK96; BP89] twice. Subsequently, the resulting square matrix is inverted and transposed. By Theorem 3.10, this procedure runs in polynomial time and has no error probability. Due to the choice of δ and the fact that $\|\tilde{G}\|_\infty \leq \sqrt{m}a$ and $\|\tilde{G}-G\| < \delta\cdot\lambda_1^*$, we can apply Theorem 3.10 to obtain $\|\Delta_B\|_\infty = \|B-\tilde{B}\| < \tau$, as required. Note that the size of δ is chosen in such a way that the decline in precision (see Theorem 3.10) is taken care of. By Theorem 3.3, the matrix $\tilde{G}$ indeed approximates a full generating set of Λ^* with overwhelming probability; implying that the output matrix $\tilde{B}$

Algorithm 1: Quantum algorithm that solves the continuous hidden subgroup problem

```
Require:
    - An $(a, r, \epsilon)$-oracle $\mathbf{f} : \mathbb{R}^m \to \mathcal{H}$ that is periodic with respect to the full-rank hidden lattice $\Lambda \subseteq \mathbb{R}^m$, whose dual lattice $\Lambda^*$ has first minimum $\lambda_1^* = \lambda_1(\Lambda^*)$. We require the parameters $\epsilon$ and $r$ to satisfy $\epsilon < 1/4$ and $r \leq \lambda_1(\Lambda)/6$.
    - An error parameter $\tau$ quantifying the maximum allowed deviation of the output basis $\tilde{B}$ from an actual basis $B$ of $\Lambda$.
Ensure: With constant probability, an $\tau$-approximated basis $\tilde{B}$ of the lattice $\Lambda$. In other words, a matrix $\tilde{B} \in \mathrm{Mat}_{m \times m}(\mathbb{Q})$ satisfying $\|\tilde{B} - B\| < \tau$ for some basis $B \in \mathrm{Mat}_{m \times m}(\mathbb{R})$ of $\Lambda$, i.e., $\tau$-close in the maximum norm induced matrix norm.
Apply the dual sampling algorithm (Algorithm 2) $k$ times, with failure probability $\eta = 1/k^2$, Gaussian deviation $s = O(\sqrt{m \log(\eta^{-1})})$ and $V = O(\frac{m \log(\eta^{-1})}{\delta \lambda_1^*})$, where $k = O(m \log[\sqrt{m} \cdot a \cdot \det(\Lambda)^{1/m}])$ and $\delta = 2^{-O(mk)} \cdot (\sqrt{m} \cdot a)^{-(m+1)} \cdot \det(\Lambda)^{-1} \cdot (\lambda_1^*)^2 \cdot \tau$.
Assemble the $k$ samples from above algorithm into a matrix $\tilde{G}$, apply the Buchmann-Pohst algorithm twice (see Section 3.7), and invert and transpose the resulting basis, yielding a matrix $\tilde{B}$.
return $\tilde{B}$.
```

approximates a basis of Λ with overwhelming probability (and not a basis of a strict *sub*lattice of Λ).

Complexity estimate. We focus first on the less important complexity, the classical complexity. This complexity is mainly driven by LLL-algorithm and inversion in step (2) of Algorithm 1. This complexity can be bounded polynomially in the dimensions and the entry sizes of the matrix involved. The dimensions of $\tilde{G}$ are $k \times m$, and can therefore by polynomially bounded by m, $\log a$ and $\log(\det \Lambda)$. The entry sizes (taking a precision of at least δ into account) can be polynomially bounded by m, $\log(\det \Lambda)$, $\log(\tau)$ and $\log(1/\lambda_1^*)$. As $\log(\det \Lambda) \leq O(m \log(1/\lambda_1^*))$ we can just omit $\log(\det \Lambda)$. Making all quantities homogeneous with respect to lattice scaling, we obtain a classical complexity of $\mathsf{poly}(m, \log \frac{a}{\lambda_1^*}, \log \frac{a}{\tau})$ bit operations.

The quantum complexity is driven by the Fourier transform in the dual lattice sampling and the Gaussian preparation step. Repeating the dual lattice sampling k times costs $O(kmQ \log(mQ))$ quantum gates and $O(mQ + n)$ qubits, where n is the number qubits required to store the values $|\mathbf{f}(x)\rangle$ of the quantum oracle in (see Theorem 3.7). Repeating k times the preparation of the Gaussian initial quantum state (within total variation distance $\eta = 1/k^2$) requires $O(kmQ \log(kmQ)^2)$ quantum gates and $O(mQ + \log(k)) = O(mQ)$ qubits (where we hide $\log(k)$ into $O(mQ)$), see Theorem 3.12. As discussed in Remark 3.13, the quantum gate complexity is slightly dominated by that of the Gaussian preparation step that occurs in Step 1 of Algorithm 2; it is $O(kmQ \log(kmQ)^2)$. The overall qubit complexity is $O(mQ + n)$.

For the estimation of the number of qubits Q needed ‘per dimension’, i.e., to prove Equation (3.24), we instantiate $\eta = 1/k^2$ and $\delta = 2^{-O(mk)} \cdot (\sqrt{m} \cdot a)^{-(m+1)} \cdot \det(\Lambda)^{-1} \cdot (\lambda_1^*)^2 \cdot \tau$ in Theorem 3.7 to obtain

$$\log(1/\delta) = (m+1)\log(\sqrt{m}a) + \log(\det(\Lambda)) + O(mk) - \log \tau - 2\log(\lambda_1^*).$$

Noting that $m \log(\sqrt{m}a) + \log(\det(\Lambda)) \in O(k) \subseteq O(mk)$, we see that

$$O\left(\log \frac{a}{\eta \cdot \delta \lambda_1^*}\right) = O(mk) + O\left(\log \frac{1}{\lambda_1^* \cdot \tau}\right) + O(\log(a/\lambda_1^*))$$

Putting $O(m \log m)$ into $O(mk)$ in Equation (3.27) yields

$$Q = O(mk) + O\left(\log \frac{a}{\lambda_1^*}\right) + O\left(\log \frac{1}{\lambda_1^* \cdot \tau}\right), \tag{3.28}$$

□

3.4. Dual Lattice Sampling Algorithm

3.4.1. The Algorithm

Given a Λ-periodic function $\mathbf{f} : \mathbb{R}^m \to \mathcal{S}$ as discussed in Section 3.3, which maps a classical input x to a quantum state $|\mathbf{f}(x)\rangle$, we consider the following quantum algorithm (see Algorithm 2, or more graphically, Figure 3.4). The algorithm has oracle access to $\mathbf{f}$, meaning that it has access to a unitary that maps $|x\rangle|0\rangle$ to $|x\rangle|\mathbf{f}(x)\rangle$. As a matter of fact, we may assume the algorithm to have oracle access to a unitary that maps $|x\rangle|0\rangle$ to $|x\rangle|\mathbf{f}(Vx)\rangle$ for a parameter $V \in \mathbb{R}$ chosen by the algorithm. Per se, x may be arbitrary in $\mathbb{R}^m$; for any concrete algorithm it is of course necessary to restrict x to some finite subset of $\mathbb{R}^m$.

The algorithm we consider follows the blueprint of the standard hidden-subgroup algorithm. Notable differences are that we need to discretize (and finitize) the continuous domain $\mathbb{R}^m$ of the function, and the algorithm starts off with a superposition that is not uniform but follows a (discretized and finitized) Gaussian distribution. The reason for the latter choice is that Gaussian distributions decay very fast and behave nicely under the Fourier transform (as they are eigenfunctions of the Fourier transform).

The algorithm is given in Algorithm 2. It uses two quantum registers, each one consisting of a certain number of qubits. Associated to the first register are *grid points*: orthonormal bases $\{|x\rangle_{\mathbb{D}^m}\}_{x \in \mathbb{D}^m}$ and $\{|y\rangle_{\hat{\mathbb{D}}^m}\}_{y \in \hat{\mathbb{D}}^m}$ where the basis vectors are labeled by $x \in \mathbb{D}^m$ and $y \in \hat{\mathbb{D}}^m$, respectively, which we identify with elements $x \in \mathbb{D}^m_{\text{rep}}$ and $y \in \hat{\mathbb{D}}^m_{\text{rep}}$ (see Section 2.2.1). The second

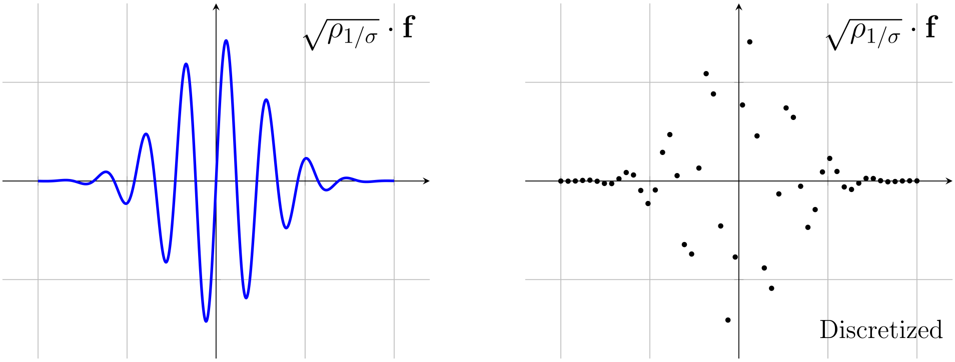

Figure 3.3.: Intuitively, it is easier to see the (quasi-)periodicity of the continuous signal (left) than that of the discrete signal (right). It is exactly the loss of information ‘between the sampling points’ that causes this chapter’s quantum algorithm to behave slightly erroneously or noisily. Of course, increasing the number of sampling points should reduce this noise; but it also causes the algorithm to need more expensive qubits. The analysis sought to keep the required qubits as low as possible, while still maintaining an acceptable error probability.

register has state space $\mathcal{H}$. The algorithm is parameterized by $q \in \mathbb{N}$ (which determines $\mathbb{D}^m$), $s > 0$ and $V > 0$.

Intuitively, the fraction $\frac{s}{V}$ is tightly related to the absolute precision of the output, whereas $\log q$ is connected with the number of qubits needed. In Algorithm 2, all quantum states described are *unnormalized* (i.e., do not have norm 1) but have all the same norm, due to the unitary operations in each step. In the analysis later, we show that, for adequately chosen parameters, the initial state $|\psi_\circ\rangle$, and therefore all states, are actually very *close* to normalized.

The description and analysis of Step 1 of Algorithm 2 is deferred to Appendix A.5. It will be shown (as summarized in Theorem 3.12) that its cost is comparable to the main cost of Algorithm 2, while contributing an error of at most $o(\eta)$ in the trace distance.

Algorithm 2: Quantum algorithm for the dual lattice sampling problem

```
**Prepare the Gaussian state**
    $|\psi_\circ\rangle := s^{m/2} \cdot \sum_{x \in \mathbb{D}^m} \sqrt{\rho_{1/s}(x)} \cdot |x\rangle_{\mathbb{D}^m} |0\rangle$ ;
**Apply the f-oracle**, yielding $s^{m/2} \cdot \sum_{x \in \mathbb{D}^m} \sqrt{\rho_{1/s}(x)} \cdot |x\rangle_{\mathbb{D}^m} |\mathbf{f}(Vx)\rangle$ ;
**Apply the quantum Fourier transform on the first register**,
    yielding the unnormalized state
    $s^{m/2} \cdot \sum_{x \in \mathbb{D}^m} \sum_{y \in \hat{\mathbb{D}}^m} \sqrt{\rho_{1/s}(x)} \cdot e^{-2\pi i \langle x, y \rangle} \cdot |y\rangle_{\hat{\mathbb{D}}^m} |\mathbf{f}(Vx)\rangle$ ;
**Measure the first register in the $\hat{\mathbb{D}}^m_{\mathrm{rep}}$-basis** yielding some
    $y \in \hat{\mathbb{D}}^m_{\mathrm{rep}}$, and output $\frac{y}{V}$ ;
```

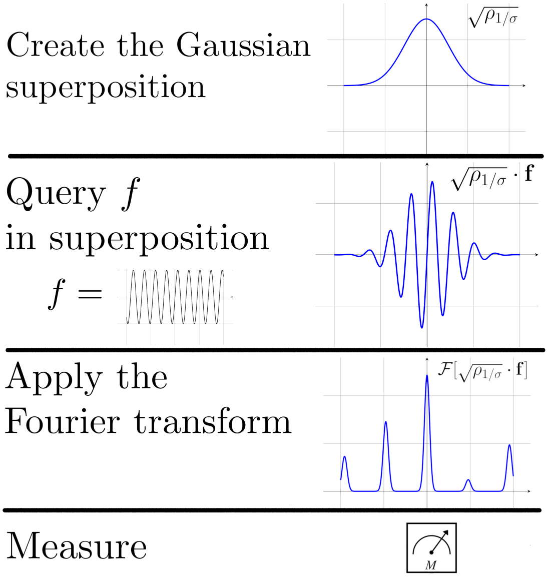
Create the Gaussian superposition
$\sqrt{\rho_{1/\sigma}}$
Query f in superposition
$f =$
$\sqrt{\rho_{1/\sigma}} \cdot \mathbf{f}$
Apply the Fourier transform
$\mathcal{F}[\sqrt{\rho_{1/\sigma}} \cdot \mathbf{f}]$
Measure
M

Figure 3.4.: A visual representation of Algorithm 2, if it would have been run on a 'continuous' quantum computer with infinitely many qubits. In reality, quantum computers have only finitely many qubits, leading to discretization errors. These errors are the main topic of this chapter. Note that the state after the Fourier transform 'peaks' at the dual lattice points.

3.4.2. The Figure of Merit

Recall that $N = \dim \mathcal{H} = 2^n$. Then the state after step (2) of Algorithm 2 equals, up to normalization,

$$|\psi\rangle := s^{m/2} \sum_{x \in \mathbb{D}^m} \sqrt{\rho_{1/s}(x)}\, |x\rangle_{\mathbb{D}^m} |\mathbf{f}(Vx)\rangle$$

which we can rewrite as

$$|\psi\rangle = \sum_{x \in \mathbb{D}^m} |x\rangle_{\mathbb{D}^m} |\mathbf{h}(x)\rangle$$

where

$$|\mathbf{h}(x)\rangle := s^{m/2} \sqrt{\rho_{1/s}(x)} \cdot |\mathbf{f}(Vx)\rangle \,.$$

Applying the quantum Fourier transform in step (3) maps this to

$$\begin{aligned} |\hat{\psi}\rangle &= q^{-m/2} \sum_{x \in \mathbb{D}^m} \sum_{y \in \hat{\mathbb{D}}^m} e^{-2\pi i \langle x, y\rangle} |y\rangle_{\hat{\mathbb{D}}^m} |\mathbf{h}(x)\rangle \\ &= q^{m/2} \sum_{y \in \hat{\mathbb{D}}^m} |y\rangle_{\hat{\mathbb{D}}^m} |\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}(y)\rangle \,, \end{aligned}$$

where the factor $q^{m/2}$ comes from the fact that, by our convention, the Fourier transform $\mathcal{F}_{\mathbb{D}^m}$ is scaled with the factor q^{-m}, while the quantum Fourier transform comes with a scaling factor $q^{-m/2}$.

Up to normalization, the probability to observe outcome $y \in \hat{\mathbb{D}}^m$ in step (4) thus is

$$\langle\hat{\psi}|(|y\rangle\langle y| \otimes \mathbb{I})|\hat{\psi}\rangle = q^m \cdot \|\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}(y)\|_{\mathcal{H}}^2 \,,$$

and so, for any "target" subset $C \subset \hat{\mathbb{D}}^m$, the probability for the algorithm to produce an outcome $y \in C$ equals

$$\mathcal{D}(C) = \sum_{y \in C} \frac{\langle\hat{\psi}|(|y\rangle\langle y| \otimes \mathbb{I})|\hat{\psi}\rangle}{\langle\psi_\circ|\psi_\circ\rangle} = \frac{\|1_C \cdot \mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}\|_{\hat{\mathbb{D}}^m}^2}{\frac{s^m}{q^m}\sum_{x \in \mathbb{D}^m} \rho_{1/s}(x)} \,. \tag{3.29}$$

This target set are the points that one *would like to have as an outcome* after measuring. In our situation, this target set C consists of points close to dual lattice points ℓ^*, as those are considered 'good' measurement (see Figure 3.5).

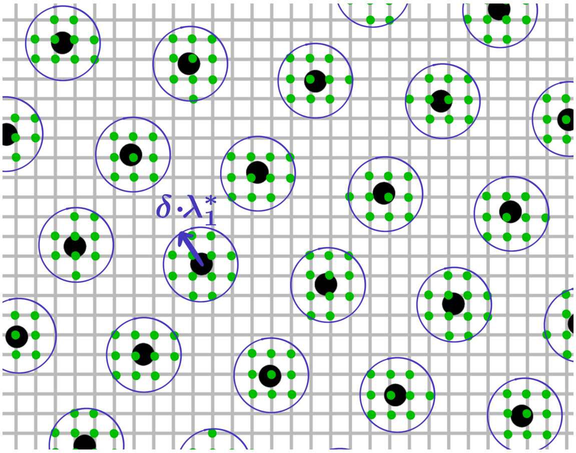

Figure 3.5.: The target set C consists of those grid points that are $\delta \cdot \lambda_1^*$-close to the dual lattice Λ^*; these points give valuable information about the dual lattice Λ^*. In this specific example, the target set consists of the green points and the blue circles around the black dual lattice points have radius $\delta \cdot \lambda_1^*$.

The algorithm's behavior in the limit

Intuitively, in the limit $q \to \infty$, the grid $\frac{1}{q}\mathbb{Z}^m$ becomes $\mathbb{R}^m$; thus, neglecting constant factors, the function $\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}$ is expected to converge to

$$\mathcal{F}_{\mathbb{R}^m}\{\rho_{\sqrt{2}/s} \cdot \mathbf{f}(V\cdot)\} = \rho_{s/\sqrt{2}} \star \mathcal{F}_{\mathbb{R}^m}\{\mathbf{f}(V\cdot)\}\,.$$

Furthermore, when V is large enough compared to s, then, relative to the dual lattice $V\Lambda^*$, the Gaussian function behaves as a Dirac delta function. Thus, the above function is then supported by $V\Lambda^*$ and takes on the values $|c_{\ell^*}\rangle$. Hence, by taking square norms, we get the claimed $\langle c_{\ell^*}|c_{\ell^*}\rangle$.

Below, we prove that this intuition is indeed correct, and we work out the actual "rate of convergence".

3.5. Analysis

3.5.1. Proof Overview

In the following few paragraphs we give an overview of the proof of correctness of Algorithm 2. The main idea boils down to showing that the finite Fourier transform is close to the continuous Fourier transform on the function $\mathbf{h} = \mathbf{f} \cdot \rho_{1/s}$. They are indeed close due to the smoothness of the Gaussian and the Lipschitz-continuity of the oracle function $\mathbf{f}$.

The unnormalized initial state $|\psi_\circ\rangle$ has approximately norm one. By the smoothing argument of Banaszczyk, we derive that the initial state's norm satisfies $\langle\psi_\circ|\psi_\circ\rangle = \frac{s^m}{q^m}\sum_{x\in\mathbb{D}^m}\rho_{1/s}(x) \approx 1$. So, the initial state might not be *perfectly* normalized, but it is almost. Therefore,

$$\mathcal{D}(C) = \frac{\|1_C \cdot \mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}\|^2_{\hat{\mathbb{D}}^m}}{\frac{s^m}{q^m}\sum_{x\in\mathbb{D}^m}\rho_{1/s}(x)} \approx \|1_C \cdot \mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}\|^2_{\hat{\mathbb{D}}^m}$$

meaning that we can focus on the latter quantity, that consists just of the norm of the Fourier transformed function $\mathbf{h}$.

Replacing the function $\mathbf{h}$ by its $\mathbb{T}^m$-periodization $\mathbf{h}|^{\mathbb{T}^m}$. The function $\mathbf{h} = s^{m/2} \cdot \mathbf{f} \cdot \rho_{\sqrt{2}/s}$ is a product of the function $\mathbf{f}$ and a Gaussian that is narrow enough to be contained within the centered unit cube. Therefore, periodization of $\mathbf{h}$ with respect to the unit cube $[-\frac{1}{2}, \frac{1}{2}]^m$ (i.e., the central representative of the unit torus) doesn't differ too much from restricting $\mathbf{h}$ to the torus. Therefore,

$$\|1_C \cdot \mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}\|^2_{\hat{\mathbb{D}}^m} \approx \|1_C \cdot \mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|^2_{\hat{\mathbb{D}}^m}.$$

Replacing the finite $\mathbb{D}^m$-Fourier transform by the $\mathbb{T}^m$-Fourier transform. Because the function $\mathbf{h}$ is Lipschitz-continuous, changing the finite Fourier transform into a continuous one over the torus $\mathbb{T}^m$ gives us a error that depends mainly on the discretization parameter q and the Lipschitz constant $\mathrm{Lip}(\mathbf{f})$.

$$\|1_C \cdot \mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|^2_{\hat{\mathbb{D}}^m} \approx \|1_C \cdot \mathcal{F}_{\mathbb{T}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|^2_{\mathbb{Z}^m}.$$

Replacing the $\mathbb{T}^m$-Fourier transform by the $\mathbb{R}^m$-Fourier transform. Using the Poisson summation formula, one can derive an equality between the Fourier transform of $\mathbf{h}|^{\mathbb{T}^m}$ over the torus $\mathbb{T}^m$ and the Fourier transform of $\mathbf{h}$ over the reals $\mathbb{R}^m$.

$$\|1_C \cdot \mathcal{F}_{\mathbb{T}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|^2_{\mathbb{Z}^m} = \|1_C \cdot \mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}\|^2_{\mathbb{Z}^m}.$$

Relating the $\mathbb{R}^m$-Fourier transform with the Fourier coefficients $|c_{\ell^*}\rangle$ of $|\mathbf{f}\rangle$. As $\mathbf{h}$ is essentially a product of $\mathbf{f}$ and a relatively wide Gaussian, one can apply the convolution theorem to obtain the real Fourier transform of $\mathbf{h}$. This Fourier transform is then very much related with the Fourier coefficients $|c_{\ell^*}\rangle$ of $\mathbf{f}$.

$$\|1_C \cdot \mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}\|^2_{\mathbb{Z}^m} \approx \sum_{\ell^* \in \Lambda^*} \langle c_{\ell^*}|c_{\ell^*}\rangle \cdot \iota_C(\ell^*)$$

The function ι_C here acts as sort-of an indicator function; one can think of $\iota_C(\ell^*)$ being close to one whenever ℓ^* is in the 'target set' C and zero otherwise. Recall that this target set are the 'wanted' points, i.e., the desired outcomes after measuring the quantum state. In our situation, this target set C consists of points close $\delta\lambda_1^*$-close to dual lattice points ℓ^*, as those are considered 'good' measurements; they namely give valuable information about the dual lattice Λ^*.

Lower bounding the success probability by means of Fourier coefficients of $\mathbf{f}$. In particular, one can show that, up to a small error, the function ι_C indeed acts as an indicator function. Whenever a large enough ball around a dual lattice point ℓ^* is contained in C, the value of $\iota_C(\ell^*)$ approximates one.

$$\mathcal{D}(C) \approx \sum_{\ell^* \in \Lambda^*} \langle c_{\ell^*}|c_{\ell^*}\rangle \cdot \iota_C(\ell^*) \geq \sum_{\substack{\ell^* \in \Lambda^* \\ \mathcal{B}_{\delta\lambda_1^*}(\ell^*) \cap \mathbb{Z}^m \subseteq C}} \langle c_{\ell^*}|c_{\ell^*}\rangle. \tag{3.30}$$

Taking into account the bounded output of Algorithm 2 and finalizing the analysis. The output distribution $\mathcal{D}$ of Algorithm 2 has support only in $[-q/2, q/2]^m$. So, for any $S \subseteq \Lambda^*$ the probability p_S from Problem 3.6 applied to the output distribution of Algorithm 2 satisfies

$$\begin{aligned} p_S = \mathcal{D}(\mathcal{B}_{\delta\lambda_1^*}(S)) &= \mathcal{D}\Big(\mathcal{B}_{\delta\lambda_1^*}(S) \cap [-q/2, q/2]^m\Big) \\ &\gtrapprox \sum_{\substack{\ell^* \in \Lambda^* \\ \ell^* \in S \cap [-q/2, q/2]^m}} \langle c_{\ell^*} | c_{\ell^*} \rangle \gtrapprox \sum_{\ell^* \in S} \langle c_{\ell^*} | c_{\ell^*} \rangle \end{aligned}$$

where the first ‘approximate inequality’ (which is an inequality up to some small error) is obtained from Equation (3.30) and the last ‘approximate inequality’ holds by the fact that the ‘tail’ of the Fourier coefficients of $\mathbf{f}$ has small weight, i.e., $\sum_{|\ell^*| > q/2} \langle c_{\ell^*} | c_{\ell^*} \rangle$ is small.

Summarizing, this error mainly occurs because of the phrasing of the Problem 3.6. It makes the suggestion that the distribution $\mathcal{D}$ should have unbounded support and should be able to reach any dual lattice point, whereas in reality (for the output distribution of Algorithm 2) this is very much not the case. The error induced by this discrepancy is, as a consequence, essentially the combined weight (i.e., the ‘lost probability’) of the lattice points unreachable by the output distribution of Algorithm 2.

The velocity parameter V. In the formal analysis below, we sometimes temporarily assume that the velocity parameter equals one, i.e., $V = 1$. This is for sake of clarity and can be done without loss of generality, since for arbitrary V the very same reasoning can be applied to the function $\mathbf{f}_V := \mathbf{f}(V\cdot)$. This affects the quantities involved in the sense that Λ^* becomes $V\Lambda^*$, λ_1^* becomes $V \cdot \lambda_1^*$ and $\operatorname{Lip}(\mathbf{f}_V)$ becomes $V \operatorname{Lip}(\mathbf{f})$.

To be clear, the end results and errors involved are always stated *for general* V. Moreover, whenever the assumption $V = 1$ occurs in a proof or a line of reasoning, we will always explicitly say so, in order to avoid confusion.

3.5.2. Formal Analysis

The unnormalized initial state $|\psi_\circ\rangle$ has approximately norm one

By the smoothing lemma (see Lemma 2.31), we have, whenever $q/s \geq \sqrt{m}$,

$$\begin{aligned}\langle\psi_\circ|\psi_\circ\rangle = \frac{s^m}{q^m}\sum_{x\in\mathbb{D}^m}\rho_{1/s}(x) \leq \frac{s^m}{q^m}\cdot\rho_{1/s}\left(\frac{1}{q}\mathbb{Z}^m\right) \leq 1+2\beta_{q/s}\\ \leq 1+O(e^{-q^2/s^2}).\end{aligned}$$

Therefore,

$$\left|\frac{\|1_C\cdot\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}\|^2_{\hat{\mathbb{D}}^m}}{\frac{s^m}{q^m}\sum_{x\in\mathbb{D}^m}\rho_{1/s}(x)} - \|1_C\cdot\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}\|^2_{\hat{\mathbb{D}}^m}\right| \leq O(e^{-q^2/s^2}). \tag{3.31}$$

By requiring that $q/s \geq \sqrt{m+\log(\eta^{-1})}$, we can safely neglect this error.

Replacing the function h by its $\mathbb{T}^m$-periodization $\mathrm{h}|^{\mathbb{T}^m}$

By the linearity of the Fourier transform, by the fact that 1_C is an indicator function and by Parseval's theorem, one can deduce

$$\begin{aligned}\|1_C\cdot\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\} - 1_C\cdot\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|_{\hat{\mathbb{D}}^m} &\leq \|\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}-\mathbf{h}|^{\mathbb{T}^m}\}\|_{\hat{\mathbb{D}}^m}\\ &= \|\mathbf{h}|^{\mathbb{T}^m}-\mathbf{h}\|_{\mathbb{D}^m}.\end{aligned}$$

Writing out the definition of the functions $\mathbf{h} = s^{m/2}\cdot\mathbf{f}\cdot\rho_{\sqrt{2}/s}$ and $\mathbf{h}|^{\mathbb{T}^m} = \sum_{z\in\mathbb{Z}^m}\mathbf{h}(z+\cdot)$, we obtain

$$\begin{aligned}\|\mathbf{h}|^{\mathbb{T}^m}-\mathbf{h}\|^2_{\mathbb{D}^m} &= \frac{1}{q^m}\sum_{x\in\mathbb{D}^m}\left\|\sum_{z\in\mathbb{Z}^m\backslash 0}\mathbf{h}(x+z)\right\|^2_{\mathcal{H}}\\ &\leq \frac{s^m}{q^m}\sum_{x\in\mathbb{D}^m}\left(\sum_{z\in\mathbb{Z}^m\backslash 0}\rho_{\sqrt{2}/s}(x+z)\cdot\|\mathbf{f}(V(x+z))\|_{\mathcal{H}}\right)^2.\end{aligned}$$

Since $\|\mathbf{f}(x)\|_{\mathcal{H}} = \sqrt{\langle \mathbf{f}(x)|\mathbf{f}(x)\rangle} = 1$, as $|\mathbf{f}(x)\rangle$ is a quantum state for any $x \in \mathbb{R}^m$, above expression is bounded by

$$\frac{s^m}{q^m}\sum_{x\in\mathbb{D}^m}\Bigg(\underbrace{\sum_{z\in\mathbb{Z}^m\backslash 0}\rho_{\sqrt{2}/s}(x+z)}_{\leq 2\cdot\beta_{\frac{s}{2\sqrt{2}}}}\Bigg)^2 \leq \frac{s^m\cdot|\mathbb{D}^m|}{q^m}\cdot(2\cdot\beta_{\frac{s}{2\sqrt{2}}})^2$$

$$\leq 4\cdot s^m\cdot(\beta_{\frac{s}{2\sqrt{2}}})^2,$$

as $\rho_{\sqrt{2}/s}(\mathbb{Z}^m\backslash\{0\}+x) \leq 2\cdot\beta_{\frac{s}{2\sqrt{2}}}$, from Banaszczyk's tail bound in Corollary 2.30. By the reverse triangle inequality, provided that $s \geq \sqrt{8m}$, we conclude

$$\Big|\|1_C\cdot\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}\}\|^2_{\hat{\mathbb{D}}^m} - \|1_C\cdot\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|^2_{\hat{\mathbb{D}}^m}\Big| \leq O(s^m e^{-s^2/8}). \tag{3.32}$$

By requiring that $s \geq \sqrt{8m\log(m)+\log(\eta^{-1})}$, we can safely neglect this error.

Replacing the finite $\mathbb{D}^m$-Fourier transform by the $\mathbb{T}^m$-Fourier transform

Using Theorem 2.8 with $\mathbf{h}|^{\mathbb{T}^m}$, one obtains

$$\Big|\|1_C\cdot\mathcal{F}_{\mathbb{D}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|_{\hat{\mathbb{D}}^m} - \|1_C\cdot\mathcal{F}_{\mathbb{T}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|_{\mathbb{Z}^m}\Big| \leq \frac{4\pi\sqrt{m}\,\mathrm{Lip}(\mathbf{h}|^{\mathbb{T}^m})}{q} \tag{3.33}$$

$$\leq O\left(\frac{\sqrt{m}s^{m/2}(V\,\mathrm{Lip}(\mathbf{f})+s^2)}{q}\right). \tag{3.34}$$

Remark 3.14. *In above inequality the indicator function 1_C is used as a function on both $\mathbb{D}^m$ and $\mathbb{Z}^m$. The function 1_C on $\mathbb{Z}^m$ must be interpreted as having the same values on $\mathbb{D}^m_{\text{rep}} \subseteq \mathbb{Z}^m$ as on $\mathbb{D}^m$ and having value zero otherwise.*

Lemma 3.15. *Assume that $s \geq 4\sqrt{m}$. Then, for the Lipschitz constant* $\operatorname{Lip}(\mathbf{h}|^{\mathbb{T}^m})$ *of* $\mathbf{h}|^{\mathbb{T}^m}$ *holds*

$$\operatorname{Lip}(\mathbf{h}|^{\mathbb{T}^m}) \leq s^{m/2}\left(2V \operatorname{Lip}(\mathbf{f}) + \pi s^2\right).$$

Proof. For the sake of clarity, we assume $V = 1$ throughout this proof; at the end we will then have to replace $\operatorname{Lip}(\mathbf{f})$ by $V \operatorname{Lip}(\mathbf{f})$. Also, we will temporarily omit the constant term $s^{m/2}$ in the definition of $\mathbf{h}$ and use ρ for $\rho_{\sqrt{2}/s}$; thus calculating with $\mathbf{h} = \mathbf{f} \cdot \rho$ instead. In the final step, the multiplicative term $s^{m/2}$ will then be multiplied again to the end result.

By applying the triangle inequality multiple times, using the fact that $\|\mathbf{f}(x)\|_{\mathcal{H}} = 1$ for all $x \in \mathbb{R}^m$ and using the Lipschitz-continuity of $\mathbf{f}$, one obtains, for every $x, y \in \mathbb{R}^m$,

$$\begin{aligned}\|\mathbf{h}(x) - \mathbf{h}(y)\|_{\mathcal{H}} &\leq \left\|\mathbf{f}(x)(\rho(x) - \rho(y))\right\|_{\mathcal{H}} + \left\|(\mathbf{f}(x) - \mathbf{f}(y))\rho(y)\right\|_{\mathcal{H}} \\ &\leq |\rho(x) - \rho(y)| + \operatorname{Lip}(\mathbf{f}) \cdot \|x - y\|_{\mathbb{R}^m} \cdot \rho(y) \qquad (3.35)\end{aligned}$$

By periodizing with respect to the unit torus $\mathbb{T}^m = \mathbb{R}^m/\mathbb{Z}^m$ and applying the triangle inequality, we obtain, for all $x, y \in [-1/2, 1/2]^m$,

$$\begin{aligned}\|\mathbf{h}|^{\mathbb{T}^m}(x) - \mathbf{h}|^{\mathbb{T}^m}(y)\|_{\mathcal{H}} \leq &\sum_{z \in \mathbb{Z}^m} |\rho(x+z) - \rho(y+z)| \\ &+ \operatorname{Lip}(\mathbf{f}) \cdot \|x - y\|_{\mathbb{T}^m} \cdot \sum_{z \in \mathbb{Z}^m} \rho(y+z) \qquad (3.36)\end{aligned}$$

By smoothing arguments of Banaszczyk, one deduces that $\rho_{\sqrt{2}/s}(y + \mathbb{Z}^m) \leq 2$ (see Corollary 2.30), where we use the assumption $s \geq 4\sqrt{m}$. By the reasoning in Lemma A.33, we have that

$$\begin{aligned}&\sum_{z \in \mathbb{Z}^m} |\rho_{\sqrt{2}/s}(x+z) - \rho_{\sqrt{2}/s}(y+z)| \\ \leq &\pi s^2/2 \cdot \|x - y\|_{\mathbb{T}^m} \underbrace{\sum_{z \in \mathbb{Z}^m} \rho_{\sqrt{8}/s}(x + y + 2z)\|x + y + 2z\|}_{\leq 2} \\ \leq &\pi s^2 \cdot \|x - y\|_{\mathbb{T}^m}, \qquad (3.37)\end{aligned}$$

where the last inequality can be obtained by absorbing $\|x+y+2z\|$ into the Gaussian and applying smoothing arguments again; $\rho_{1/s}(x)\cdot\|x\| \leq \rho_{2/s}(x)$ for all $x \in \mathbb{R}^m$ and $s \geq \sqrt{m}$, and $\rho_{\sqrt{8}/s}(\mathbb{Z}^m) \leq \rho_{\sqrt{m}}(\mathbb{Z}^m) \leq 1 + 2\cdot\beta_{\sqrt{m}} \leq 2$, for $s \geq 4\sqrt{m}$ (see Lemma 2.29). In other words,

$$\begin{aligned}\sum_{z\in\mathbb{Z}^m} \rho_{\sqrt{8}/s}(x+y+2z)\|x+y+2z\| &\leq \sum_{z\in\mathbb{Z}^m} \rho_{\sqrt{32}/s}(x+y+2z)\\ &\leq \rho_{\sqrt{32}/s}(2\cdot\mathbb{Z}^m) = \rho_{\sqrt{8}/s}(\mathbb{Z}^m) \leq 2.\end{aligned}$$

By combing Equations (3.35) to (3.37), multiplying the factor $s^{m/2}$ and replacing $\mathrm{Lip}(\mathbf{f})$ by $V\cdot\mathrm{Lip}(\mathbf{f})$ we obtain the final result. ☐

Replacing the $\mathbb{T}^m$-Fourier transform by the $\mathbb{R}^m$-Fourier transform

Apply the Poisson summation formula (see Corollary 2.5) to conclude that

$$\|1_C \cdot \mathcal{F}_{\mathbb{T}^m}\{\mathbf{h}|^{\mathbb{T}^m}\}\|_{\mathbb{Z}^m} = \|1_C \cdot \mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}\|_{\mathbb{Z}^m}\,,$$

where $\mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}$ is temporarily identified with its restriction to $\mathbb{Z}^m$.

Relating the $\mathbb{R}^m$-Fourier transform with the Fourier coefficients $|c_{\ell^*}\rangle$ of $|\mathbf{f}\rangle$

By applying the convolution theorem as outlined in Equation (2.9) of Section 2.2.2, we see that

$$\begin{aligned}\mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}[y] &= \mathcal{F}_{\mathbb{R}^m/\Lambda}\{\mathbf{f}(V\cdot)\} \star \mathcal{F}_{\mathbb{R}^m}\{s^{m/2}\rho_{\sqrt{2}/s}(\cdot)\}(y)\\ &= \left(\frac{2}{s}\right)^{m/2} \sum_{\ell^*\in\Lambda^*} |c_{\ell^*}\rangle \rho_{s/\sqrt{2}}(y - V\ell^*),\end{aligned}$$

where $|c_{\ell^*}\rangle$ are the vectorial Fourier coefficients of $\mathbf{f}$. Therefore,

$$\begin{aligned}&\|\mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}[y]\|^2_{\mathcal{H}}\\ &= \left(\frac{2}{s}\right)^m \sum_{k^*\in\Lambda^*}\sum_{\ell^*\in\Lambda^*} \langle c_{\ell^*}|c_{k^*}\rangle \rho_{s/\sqrt{2}}(y - V\ell^*)\rho_{s/\sqrt{2}}(y - Vk^*)\\ &= \left(\frac{2}{s}\right)^m \sum_{u^*\in\frac{1}{2}\Lambda^*}\sum_{v^*\in u^*+\Lambda^*} \langle c_{v^*+u^*}|c_{v^*-u^*}\rangle \rho_{s/2}(Vu^*)\rho_{s/2}(y - Vv^*)\,, \qquad (3.38)\end{aligned}$$

where the latter is obtained by the variable substitution $u^* = \frac{\ell^* - k^*}{2}$, $v^* = \frac{\ell^* + k^*}{2}$, and using the multiplicative properties of Gaussian functions (see Lemma 2.23), like $\rho_{s/\sqrt{2}}(x)\rho_{s/\sqrt{2}}(y) = \rho_{s/2}((x+y)/2)\rho_{s/2}((x-y)/2)$ for all $x, y \in \mathbb{R}^m$.

Definition 3.16. *For any subset $C \subseteq \mathbb{Z}^m$, any $s > 0$ and any $\ell^* \in \Lambda^*$, we define $\iota_C : \Lambda^* \to \mathbb{R}_{>0}$ by the following rule,*

$$\iota_C(\ell^*) := \left(\frac{2}{s}\right)^m \sum_{y \in C} \rho_{s/2}(y - V\ell^*),$$

where leave out the dependence on s and V in the notation.

The above definition of ι_C is mainly to make the notation in this analysis more compact. But this function on Λ^* also has an intuitive interpretation; it is the cumulative Gaussian weight of all points in C around ℓ^* (or, $V \cdot \ell^*$ in the case of scaling with V). So, if C contains many close points around ℓ^* (see Figure 3.5 and Figure 3.6), this cumulative Gaussian weight approaches 1, whereas if there are no points in C around ℓ^*, this weight approaches zero. Summarizing, the value $\iota_C(\ell^*)$ quantifies the number of close points around ℓ^*; a value of 1 indicates many good close points in C, whereas a value near 0 indicates no good close points (see Figure 3.6).

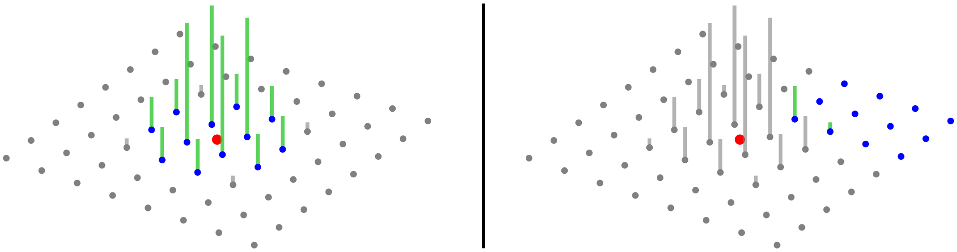

Figure 3.6.: The function $\iota_C(\ell^*)$ equals the cumulative Gaussian weight of all points in C around ℓ^*. In the left panel above, the set C contains many points around the red lattice point ℓ^*, yielding a cumulative Gaussian weight approaching one, i.e., $\iota_C(\ell^*) \approx 1$. In the right panel, set C only contains a few points close to the lattice point, yielding a very low Gaussian weight, i.e., $\iota_C(\ell^*) \approx 0$.

Lemma 3.17. *Let* $V, s > 0$ *satisfy the conditions* $V\lambda_1^*/s \geq \sqrt{m}$ *and* $s \geq \sqrt{m}$*. Then, for any* $C \subseteq [q]_c^m$*, we have*

$$\left| \|1_C \mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}\|_{\mathbb{Z}^m}^2 - \sum_{\ell^* \in \Lambda^*} \langle c_{\ell^*} | c_{\ell^*} \rangle \iota_C(\ell^*) \right| \leq O(e^{-(V\lambda_1^*/s)^2}). \tag{3.39}$$

Proof. Without loss of generality, we assume in the rest of the proof that $V = 1$, as sketched in the last paragraph of Section 3.5.1. At the end of the proof we will then replace λ_1^* by $V \cdot \lambda_1^*$.

By writing out the definition of the norm over $\mathbb{Z}^m$ and using Equation (3.38), we obtain

$$\begin{aligned}\|1_C \mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}\|_{\mathbb{Z}^m}^2 &= \sum_{y \in C} \|\mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}[y]\|_{\mathcal{H}}^2 \\ &= \left(\frac{2}{s}\right)^m \sum_{y \in C} \sum_{\substack{u^* \in \frac{1}{2}\Lambda^* \\ v^* \in u^* + \Lambda^*}} \langle c_{v^*+u^*} | c_{v^*-u^*} \rangle \rho_{s/2}(u^*) \rho_{s/2}(y - v^*)\,.\end{aligned}$$

By swapping the summation over C to the right, we deduce

$$\|1_C \mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}\|_{\mathbb{Z}^m}^2 = \sum_{\substack{u^* \in \frac{1}{2}\Lambda^* \\ v^* \in u^* + \Lambda^*}} \langle c_{v^*+u^*} | c_{v^*-u^*} \rangle \rho_{s/2}(u^*) \iota_C(v^*)\,.$$

We split above sum into a part where $u^* = 0$ and a part where $u^* \neq 0$. Notice that for the case $u^* = 0$, the inner product $\langle c_{v^*+u^*} | c_{v^*-u^*} \rangle$ becomes $\langle c_{v^*} | c_{v^*} \rangle$ and $\rho_{s/2}(u^*) = 1$. This yields

$$\begin{aligned}\|1_C \mathcal{F}_{\mathbb{R}^m}\{\mathbf{h}\}\|_{\mathbb{Z}^m}^2 = &\sum_{\ell^* \in \Lambda^*} \langle c_{\ell^*} | c_{\ell^*} \rangle \cdot \iota_C(\ell^*) \\ &+ \sum_{u^* \in \frac{1}{2}\Lambda^* \backslash 0} \rho_{s/2}(u^*) \sum_{v^* \in u^* + \Lambda^*} \langle c_{v^*+u^*} | c_{v^*-u^*} \rangle \cdot \iota_C(v^*). \end{aligned} \tag{3.40}$$

In order to achieve the claim of this lemma, it is enough to bound the second term (where $u^* \neq 0$) in Equation (3.40). As we assumed that $s \geq \sqrt{m}$, we can bound $\iota_C(v^*) \leq \left(\frac{2}{s}\right)^m \rho_{s/2}(\mathbb{Z}^m + t) \leq 2$ for any $v^* \in \mathbb{R}^m$ and $C \subseteq \mathbb{Z}^m$ by applying smoothing arguments (see Corollary 2.32). The sum of the ‘shifted inner products' of the Fourier coefficients is bounded by one, as can be seen by applying the Cauchy-Schwarz inequality and the inequality of arithmetic and geometric means.

$$\left|\sum_{v^*\in u^*+\Lambda^*}\langle c_{v^*+u^*}|c_{v^*-u^*}\rangle\right| \leq \sum_{v^*\in\Lambda^*}\sqrt{\langle c_{v^*+2u^*}|c_{v^*+2u^*}\rangle\langle c_{v^*}|c_{v^*}\rangle}$$

$$\leq \sum_{v^*\in\Lambda^*}\frac{\langle c_{v^*+2u^*}|c_{v^*+2u^*}\rangle + \langle c_{v^*}|c_{v^*}\rangle}{2} = \|\mathbf{f}\|^2_{\mathbb{R}^m/\Lambda} = 1.$$

Combining above reasoning with a tail bound of Banaszczyk (Lemma 2.29) the $u^* \neq 0$ part in Equation (3.40) can be bounded as follows.

$$\begin{aligned}
&\sum_{u^*\in\frac{1}{2}\Lambda^*\backslash 0}\rho_{s/2}(u^*)\left|\sum_{v^*\in u^*+\Lambda^*}\langle c_{v^*+u^*}|c_{v^*-u^*}\rangle\cdot\underbrace{\iota_C(v^*)}_{\leq 2}\right| \\
&\leq 2\sum_{u^*\in\frac{1}{2}\Lambda^*\backslash 0}\rho_{s/2}(u^*)\underbrace{\left|\sum_{v^*\in u^*+\Lambda^*}\langle c_{v^*+u^*}|c_{v^*-u^*}\rangle\right|}_{\leq 1} \\
&\leq 2\sum_{u^*\in\frac{1}{2}\Lambda^*\backslash 0}\rho_{s/2}(u^*) \leq 2\cdot\rho_s\left(\Lambda^*\setminus 0\right) \leq 4\cdot\beta_{\lambda_1^*/s}.
\end{aligned}$$

In order to drop the assumption that $V = 1$ from the start of the proof, we need to replace λ_1^* by $V\cdot\lambda_1^*$ in above expression. Applying the bound $4\cdot\beta_{V\lambda_1^*/s} \leq O(e^{-(V\lambda_1^*/s)^2})$ for $V\lambda_1^*/s \geq \sqrt{m}$ yields the final claim. ☐

By requiring that $V\lambda_1^*/s \geq \sqrt{m+\log(\eta^{-1})}$, we can safely neglect the error from Lemma 3.17.

Lower bounding the success probability by means of Fourier coefficients of f

Whenever $\mathcal{B}_{\delta\lambda_1^* V}(V\ell^*) \cap \mathbb{Z}^m \subseteq C$ for an $\ell^* \in \Lambda^*$, it holds that

$$\begin{aligned}\iota_C(\ell^*) &= \left(\frac{2}{s}\right)^m \sum_{y \in C} \rho_{s/2}(y - V\ell^*) \geq \left(\frac{2}{s}\right)^m \sum_{y \in \mathcal{B}_{V\delta\lambda_1^*}(V\ell^*) \cap \mathbb{Z}^m} \rho_{s/2}(y - V\ell^*) \\ &\geq \left(\frac{2}{s}\right)^m \rho_{s/2}(\mathbb{Z}^m)\left(1 - \beta_{2V\delta\lambda_1^*/s}\right) \geq (1 - 2 \cdot \beta_{s/2})(1 - \beta_{2V\delta\lambda_1^*/s}),\end{aligned}$$

where the second inequality follows from Banaszczyk's tail bound (see Lemma 2.25) and the last from the smoothing bound in Lemma 2.31. In other words, $\iota_C(\ell^*)$ is close to one if C contains all vectors in $\hat{\mathbb{D}}^m$ that are $\delta\lambda_1^* V$-close to $V\ell^*$. This coincides with the intuitive explanation after Definition 3.16. Note that $\delta\lambda_1^*$ is the maximum distance from a dual lattice point ℓ^* required to consider the output valuable.

It follows then that

$$\begin{aligned}&\Big| \sum_{\ell^* \in \Lambda^*} \langle c_{\ell^*} | c_{\ell^*} \rangle \iota_C(\ell^*) - \sum_{\substack{\ell^* \in \Lambda^* \\ \mathcal{B}_{V\delta\lambda_1^*}(V\ell^*) \cap \mathbb{Z}^m \subseteq C}} \langle c_{\ell^*} | c_{\ell^*} \rangle \Big| \\ \leq &O(e^{-s^2/4}) + O(e^{-(2V\delta\lambda_1^*/s)^2}),\end{aligned} \tag{3.41}$$

where we use the fact that $\sum_{\ell^* \in \Lambda^*} \langle c_{\ell^*} | c_{\ell^*} \rangle = \|\mathbf{f}\|^2_{\mathbb{R}^m/\Lambda} = 1$. By requiring that $\delta V \lambda_1^*/s \geq \sqrt{m + \log(\eta^{-1})}$ and $s \geq 4\sqrt{m + \log(\eta^{-1})}$, we can safely neglect this error.

Taking into account the bounded output of Algorithm 2 and finalizing the analysis

As the output distribution $\mathcal{D}$ of Algorithm 2 has support only in $[-q/2, q/2]^m$, we have, for any $S \subseteq \Lambda^*$,

$$\mathcal{D}(\mathcal{B}_{\delta\lambda_1^*}(S)) = \mathcal{D}\Big(\mathcal{B}_{\delta\lambda_1^*}(S) \cap [-q/2, q/2]^m\Big).$$

By simply splitting the set $S \subseteq \Lambda^*$ into an 'tail part' $S_{\text{tail}} = S \backslash [-q/4, q/4]^m$ and a 'bounded, finite part' $S_{\text{fin}} = S \cap [-q/4, q/4]^m$, we obtain

$$\sum_{\ell^* \in S} \langle c_{\ell^*} | c_{\ell^*} \rangle - \mathcal{D}(\mathcal{B}_{\delta\lambda_1^*}(S)) = \underbrace{\sum_{\ell^* \in S_{\text{tail}}} \langle c_{\ell^*} | c_{\ell^*} \rangle}_{\text{Small because of a tail bound}} + \underbrace{\sum_{\ell^* \in S_{\text{fin}}} \langle c_{\ell^*} | c_{\ell^*} \rangle - \mathcal{D}\Big(\mathcal{B}_{\delta\lambda_1^*}(S) \cap [-q/2, q/2]^m\Big)}_{\text{Small because of the error analysis}}. \tag{3.42}$$

By the fact that $\mathbf{f}$ is a Lipschitz continuous function, its Fourier coefficients have a tail bound. By applying Corollary 2.34 with $B = q/4$, we obtain the following bound

$$\sum_{\ell^* \in S_{\text{tail}}} \langle c_{\ell^*} | c_{\ell^*} \rangle \leq \sum_{\ell^* \in \Lambda^* \backslash [-q/4, q/4]^m} \langle c_{\ell^*} | c_{\ell^*} \rangle \leq \frac{4 \cdot \operatorname{Lip}(f)^2}{\pi^2 q^2}.$$

The summand in Equation (3.42) is, by the full error analysis, bounded by

$$\sum_{\mathcal{B}_{\delta\lambda_1^*}(\ell^*) \subseteq \mathcal{B}_{\delta\lambda_1^*}(S) \cap [-q/2, q/2]^m} \langle c_{\ell^*} | c_{\ell^*} \rangle - \mathcal{D}\Big(\mathcal{B}_{\delta\lambda_1^*}(S) \cap [-q/2, q/2]^m\Big)$$

$$\leq O\left(\frac{\sqrt{m} s^{m/2} (V \operatorname{Lip}(\mathbf{f}) + s^2)}{q}\right) + o(\eta) \tag{3.43}$$

As the only non-negligible error is caused by Equation (3.33), provided that $\delta V \lambda_1^*/s \geq \sqrt{m + \log(\eta^{-1})}$, $s \geq 4\sqrt{m \log m + \log(\eta^{-1})}$ and $q/s \geq \sqrt{m + \log(\eta^{-1})}$.

Remark 3.18. *Note that we chose for $S_{\mathit{fin}} = S \cap [-q/4, q/4]^m$ the box $[-q/4, q/4]^m$, whereas in the analysis we used the box $[-q/2, q/2]^m$. This is to crudely include also all points that are $\delta\lambda_1^*$-close to dual lattice vectors.*

Final theorem

Assembling all errors, we obtain the following theorem.

Theorem 3.7. *Algorithm 2 solves the Dual Lattice Sampling Problem with parameters η and δ; it uses one call to the Gaussian superposition subroutine (see Theorem 3.12), one quantum oracle call to $\mathbf{f}$, $mQ + n$ qubits, and $O(mQ\log(mQ))$ quantum gates, where*

$$Q = O\left(m \log\left(m\right)\right) + O\left(\log\left(\frac{a}{\eta \cdot \delta\lambda_1^*}\right)\right). \tag{3.27}$$

Proof. In Algorithm 2, two quantum registers are used: one to encode the grid $\mathbb{D}^m$ and another one for the storage of the state of the continuous hidden subgroup oracle $|\mathbf{f}(x)\rangle$. As the grid has q^m points, we need $m\log q$ qubits to encode it. For the oracle state it is assumed that it can be stored in n qubits, thus arriving at a total of $mQ + n$ qubits, where $Q = \log q$. Apart from constructing the initial Gaussian superposition, the only part of Algorithm 2 that uses quantum gates is the quantum Fourier transform on the grid register consisting of mQ qubits. Using a result of Hallgren et al., a sufficient approximation of this quantum Fourier transform can be obtained using only $O(mQ\log(mQ))$ elementary quantum gates [HH00].

To compute the value of $Q = \log(q)$, we instantiate the parameters $s = 4\sqrt{m\log m + \log(\eta^{-1})}$ and $V = \frac{4}{\delta\lambda_1^*} \cdot (m\log m + \log(\eta^{-1}))$. This implies $s \geq 4\sqrt{m\log m + \log(\eta^{-1})}$ and $\delta V\lambda_1^*/s \geq \sqrt{m + \log(\eta^{-1})}$, making the errors from Equations (3.31), (3.32), (3.39) and (3.41) all negligible compared to η. To get the errors from Equation (3.33) and Equation (3.43) well below η, we put

$$\log q = Q = O\left(m\log(s) + \log\left(\frac{V\,\mathrm{Lip}(\mathbf{f})}{\eta}\right)\right).$$

Writing out the instantiations of s and V and grouping the resulting expressions properly, we arrive at Equation (3.27). Here we use the fact that, for all $\eta > 0$ and $m \in \mathbb{N}$, $m\Big(\log\left(m\log m + \log(1/\eta)\right)\Big) \in O(m\log m + \log(1/\eta))$.

$\square$

3.6. From Sampling to Full Dual Lattice Recovery

We have so far focused on approximate sampling dual lattice points with probability weights $\langle c_{\ell^*} | c_{\ell^*} \rangle$ for $\ell^* \in \Lambda^*$, regardless of how useful this distribution may be. Indeed, until now, it could be that the function $\mathbf{f} : \mathbb{R}^m/\Lambda \to \mathcal{S}$ is constant, and therefore that all weight is concentrated on $0 \in \Lambda^*$. We would like now make sure we can reconstruct (approximately) Λ^* from such samples, i.e., that a sufficient number of sampled vectors from Λ^* will generate it. Informally, an equivalent condition is that the weight $\langle c_{\ell^*} | c_{\ell^*} \rangle$ is not concentrated on any proper sublattice $M^* \subsetneq \Lambda^*$. This is exactly what happens if the oracle function $\mathbf{f}$ is separating, i.e., is not too constant.

More formally, we give the following sufficient conditions for a distribution to be useful as a (approximate) lattice sampling distribution.

Definition 3.19. *Let $L \subseteq \mathbb{R}^m$ be a full-rank lattice. A distribution $\mathcal{D}$ on L is called p-*evenly distributed* whenever $\Pr_{v \leftarrow \mathcal{D}}[v \in L'] \leq p$ for any proper sublattice $L' \subsetneq L$.*

Definition 3.20. *Let $L \subseteq \mathbb{R}^m$ be a full-rank lattice. A distribution $\mathcal{D}$ on L is called (R, q)-*concentrated* whenever $\Pr_{v \leftarrow \mathcal{D}}[\|v\| \geq R] \leq q$.*

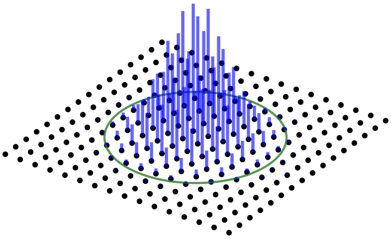

Figure 3.7.: An example of a (R, q)-concentrated distribution, where R is the radius of the green circle and $q = 0.05$, i.e., less than 5 percent of the weight lies outside the circle. Note that this Gaussian distribution is also 0.5-evenly distributed.

The following lemma states that an evenly distributed and well-concentrated distribution on a lattice L should eventually output a full generating set of

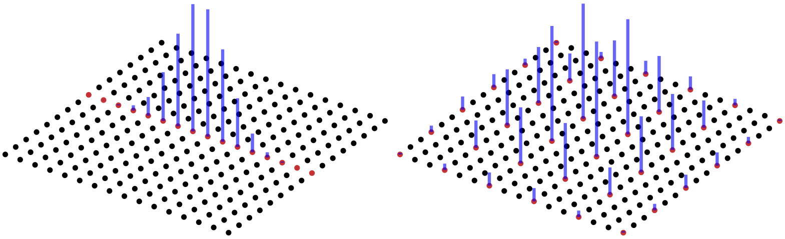

Figure 3.8.: Both these distributions are not p-evenly distributed for any $p < 1$, as the strict sublattices indicated by the red points have all of the weight.

that lattice, and gives a precise probabilistic upper bound on the number of samples needed.

Lemma 3.21. *Let $L \subseteq \mathbb{R}^m$ be a full-rank lattice with a p-evenly distributed and (R, q)-concentrated distribution $\mathcal{D}$ with $R \geq \det(L)^{1/m}$. Denote by S the random variable defined by the number of samples that needs to be drawn from $\mathcal{D}$ such that the samples together generate L as a lattice. Then, for all $\alpha > 0$,*

$$\Pr\left[S > (2+\alpha) \cdot \frac{(t+m)}{1-p-q}\right] \leq \exp(-\alpha(t+m)/2)$$

where $t = m\log_2(R) - \log_2(\det(L)) \geq 0$.

Proof. First, we define the following sublattices of L, for any $v_1, \ldots, v_{j-1} \in L$.

$$L_{v_1,\ldots,v_{j-1}} = \begin{cases} \operatorname{span}_{\mathbb{R}}(v_1, \ldots, v_{j-1}) \cap L & \text{if} \dim(\operatorname{span}_{\mathbb{R}}(v_1, \ldots, v_{j-1})) < m \\ \mathbb{Z}v_1 + \cdots + \mathbb{Z}v_{j-1} & \text{otherwise.} \end{cases}$$

Consider a sequence of samples $(v_i)_{i>0}$ (from $\mathcal{D}$). We call v_j 'good' whenever $\|v_j\| \leq R$ and $v_j \notin L_{v_1,\ldots,v_{j-1}}$. We argue that we need at most $m + t$ good vectors to generate L.

Denote L' for the lattice generated by the $m + t$ good vectors. Then the first m good vectors ensure that L' is of rank m, whereas the last t good vectors will reduce the index of the L' lattice in L. Calculating determinants, using the fact that all good vectors are bounded by R, we have $\det(L') \leq R^m/2^t \leq \det(L)$. This yields $L' = L$.

Denote by X the random variable having the negative binomial distribution with success probability $p+q$ and number of 'failures' $m+t$. That is, X is the number of independent samples from a $(p+q)$-Bernoulli distribution until $m+t$ 'failures'[1] are obtained. We argue that the random variable S is dominated by the random variable X, i.e., $\Pr[S > x] \leq \Pr[X > x]$ for every $x \in \mathbb{N}$.

Again, consider a sequence of samples $(v_i)_{i>0}$ (from $\mathcal{D}$). The probability of v_j being a 'good' vector is at least $1-p-q$, by the fact that $\mathcal{D}$ is (R,q)-concentrated and p-evenly distributed. Because at most $m+t$ 'good' vectors are needed to generate the whole lattice, S is indeed dominated by X. Therefore, for any $k \in \mathbb{N}$,

$$\begin{aligned} \Pr\left[S > \frac{t+m+k}{1-p-q}\right] &\leq \Pr\left[X > \frac{t+m+k}{1-p-q}\right] \leq \Pr\left[B < m+t\right] \\ &\leq \exp\left(-\frac{1}{2}\frac{k^2}{t+m+k}\right) \end{aligned} \tag{3.44}$$

where B is binomially distributed with $\lfloor \frac{t+m+k}{1-p-q} \rfloor$ trials and success probability $1-p-q$. The first inequality follows from the fact that S is upper bounded by X. The second inequality comes from the close relationship between the negative binomial distribution and the binomial distribution [GKP94, Ch. 8, Example 17]. The last inequality follows from the Chernoff bound. Putting $k = (1+\alpha)(t+m)$ into Equation (3.44) yields the claim. $\square$

We conclude this section by relating the parameters (a, r, ϵ) of the HSP oracle (Definition 3.2) $\mathbf{f} : \mathbb{R}^m/\Lambda \to \mathcal{S}$ to how equally-distributed and well-concentrated the distribution $\mathcal{D}_{ideal}$ on Λ^* is, arising from the Fourier coefficients of the oracle function $\mathbf{f}$. The exact relation is stated in Proposition 3.24, but we first need two technical lemmas to help us proving this relation.

---

[1] In our case, the failures are the 'good' vectors. We nonetheless chose the word 'failure' because it is standard nomenclature for the negative binomial distribution.

Lemma 3.22. *Let* Λ *be a lattice, and let* $M \supsetneq \Lambda$ *a proper super-lattice of* Λ*. Then there exists a* $v \in M$ *such that* $d(v, \Lambda) \geq \lambda_1(\Lambda)/3$*.*

Proof. Let $w \in M$ be the shortest non-zero vector in M and write $\|w\| = \alpha\lambda_1(\Lambda)$ for $\alpha \leq 1$. We consider two cases depending on the value of $\alpha \in (0,1]$. If $\alpha \geq 1/3$, choose an element $v \in M \backslash \Lambda$ arbitrarily. This element satisfies $d(v, \Lambda) \geq \lambda_1(\Lambda)/3$, since

$$\begin{aligned} d(v, \Lambda) &= d(v + \Lambda, 0) = d((v + \Lambda)\backslash 0, 0) \\ &\geq d(M\backslash 0, 0) = \alpha \cdot \lambda_1(\Lambda) \geq \lambda_1(\Lambda)/3. \end{aligned}$$

If, on the other hand, $\alpha < 1/3$, then $v = \lceil \frac{1}{3\alpha} \rceil \cdot w \in M$ satisfies $d(v, \Lambda) \geq \lambda_1(\Lambda)/3$. One can deduce this by observing that

$$\|v\| = \lceil \tfrac{1}{3\alpha} \rceil \cdot \alpha \cdot \lambda_1(\Lambda) \in [\tfrac{1}{3} \cdot \lambda_1(\Lambda), \tfrac{2}{3} \cdot \lambda_1(\Lambda)],$$

which in particular implies that $\|v - \ell\| \geq |\|\ell\| - \|v\|| \geq \frac{1}{3} \cdot \lambda_1(\Lambda)$, for all $\ell \in \Lambda$, i.e., $d(v, \Lambda) \geq \frac{1}{3}\lambda_1(\Lambda)$. ☐

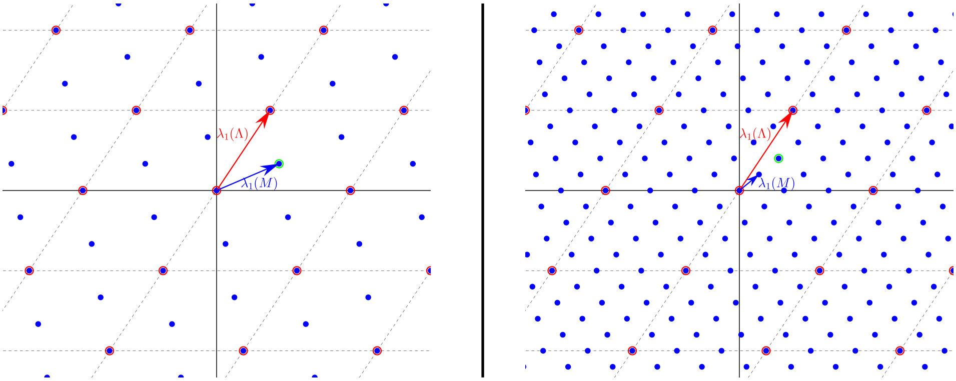

Figure 3.9.: The two cases of Lemma 3.22 are depicted here, where the smaller lattice Λ consists of the points inside the red circles. The blue super lattice M satisfies $\lambda_1(M) = \alpha\lambda_1(\Lambda)$ for some $\alpha > 1/3$ in the left picture and for some $\alpha < 1/3$ in the right picture. In both cases, an element $v \in M$ for which holds $d(v, \Lambda) > \frac{1}{3} \cdot \lambda_1(\Lambda)$ can be reasonably found. Examples of such $v \in M$ are marked with a green circle.

Lemma 3.23. *Let Λ be a lattice and $M \supsetneq \Lambda$ a proper super-lattice of Λ. Then the number $N = \left|\left\{c \in M/\Lambda \mid d(c, \Lambda) < \frac{1}{6}\lambda_1(\Lambda)\right\}\right|$ of close cosets is at most $\frac{1}{2} \cdot |M/\Lambda|$.*

Proof. By Lemma 3.22 there exists a $v \in M$ such that $d(v, \Lambda) \geq \frac{1}{3}\lambda_1(\Lambda)$. Denoting $T = \left\{c \in M/\Lambda \mid d(c, \Lambda) < \frac{1}{6}\lambda_1(\Lambda)\right\}$, we can deduce that $T \cup (T + v)$ is a disjoint union in M/Λ. Indeed, elements $c \in T$ satisfy $d(c, \Lambda) < \frac{1}{6}\lambda_1(\Lambda)$, whereas $c' \in T + v$ satisfy $d(c', \Lambda) \geq d(v, \Lambda) - \frac{1}{6}\lambda_1(\Lambda) \geq \frac{1}{6}\lambda_1(\Lambda)$. Therefore $N = |T| \leq \frac{1}{2}|M/\Lambda|$. ☐

Proposition 3.24. *Let $\mathbf{f} : \mathbb{R}^m \to \mathcal{S}$ be an (a, r, ϵ)-HSP oracle of the full-rank lattice $\Lambda \subset \mathbb{R}^m$, with $r \leq \lambda_1(\Lambda)/6$. Let $\mathcal{D}_{\mathbf{f}}$ be the distribution supported by Λ^*, with weight $\langle c_{\ell^*} | c_{\ell^*} \rangle$ at $\ell^* \in \Lambda^*$, where $|c_{\ell^*}\rangle$ are the vectorial Fourier coefficients of the function $\mathbf{f}$. Then $\mathcal{D}_{\mathbf{f}}$ is both $(\frac{1}{2} + \epsilon)$-evenly distributed and $(R, \frac{ma^2}{4\pi^2 R^2})$-concentrated for any $R > 0$.*

Proof. The distribution $\mathcal{D}_{\mathbf{f}}$ being $(R, \frac{ma^2}{4\pi^2 R^2})$-concentrated for any $R > 0$ is a direct consequence of Corollary 2.34. For the $(\frac{1}{2} + \epsilon)$-evenly distributed part, we argue as follows. Let M^* be any strict sublattice of Λ^*, and let M be its dual, which is then a superlattice of Λ. Put $\mathbf{f}|^{\mathbb{R}^m/M}(x) = \frac{1}{|M/\Lambda|} \sum_{v \in M/\Lambda} \mathbf{f}(x + v)$, the periodization of $\mathbf{f}$ with respect to $\mathbb{R}^m/M$ (c.f. Definition 2.3). We have the following sequence of equalities, of which the second follows from the Poisson summation formula (see Theorem 2.4) and the third from Parseval's theorem (see Equation (2.5)).

$$\begin{aligned}
&\sum_{v^*\in M^*} \langle c_{v^*}|c_{v^*}\rangle \\
&= \left\| \mathcal{F}_{\mathbb{R}^m/\Lambda}\{\mathbf{f}\}\Big|_{M^*} \right\|_{M^*}^2 = \left\| \mathcal{F}_{\mathbb{R}^m/M}\{\mathbf{f}|^{\mathbb{R}^m/M}\} \right\|_{M^*}^2 \\
&= \|\mathbf{f}|^{\mathbb{R}^m/M}\|_{\mathbb{R}^m/M} = \frac{1}{\det M}\int_{x\in\mathbb{R}^m/M} \langle \mathbf{f}|^{\mathbb{R}^m/M}|\mathbf{f}|^{\mathbb{R}^m/M}\rangle dx, \\
&= \frac{1}{|M/\Lambda|^2}\sum_{v,w\in M/\Lambda} \underbrace{\frac{1}{\det M}\int_{x\in\mathbb{R}^m/M} \langle \mathbf{f}(x+v)|\,\mathbf{f}(x+w)\rangle\, dx}_{I_{v,w}} \\
&= \frac{1}{|M/\Lambda|^2}\sum_{\substack{v,w\in M/\Lambda \\ d_{\mathbb{R}^m/\Lambda}(v,w)<r}} I_{v,w} + \frac{1}{|M/\Lambda|^2}\sum_{\substack{v,w\in M/\Lambda \\ d_{\mathbb{R}^m/\Lambda}(v,w)\geq r}} I_{v,w}. \qquad (3.45)
\end{aligned}$$

By the definition of an (a, r, ϵ)-oracle, we have that $|I_{v,w}| \leq \epsilon$ whenever $d_{\mathbb{R}^m/\Lambda}(v, w) \geq r$. In the rest of the cases we have $|I_{v,w}| \leq 1$, because $\mathbf{f}$ maps to the unit sphere. Equation (3.45) is therefore bounded by $\frac{|M/\Lambda\,\cap\,\mathcal{B}_r|}{|M/\Lambda|} + \epsilon$, where $\mathcal{B}_r$ is the open unit ball around zero with radius r. By Lemma 3.23, we have $\frac{|M/\Lambda\,\cap\,r\mathcal{B}|}{|M/\Lambda|} \leq \frac{1}{2}$ for $r \leq \lambda_1(\Lambda)/6$. Summarizing, we derive

$$\sum_{v^*\in M^*} \langle c_{v^*}|c_{v^*}\rangle \leq \frac{1}{2} + \epsilon.$$

Since M^* was chosen arbitrarily, we can conclude that $\mathcal{D}_{\mathbf{f}}$ is $(\frac{1}{2} + \epsilon)$-evenly distributed. □

Remark 3.25. *A similar reasoning happens in [Reg04a, Lecture 12], though it specifically targets the discrete Gaussian distribution on lattices. Despite being not general enough for our purposes, it may well be helpful for optimizing a future specialization.*

Theorem 3.9. *Let* $\mathbf{f} : \mathbb{R}^m \to \mathcal{S}$ *be an* (a, r, ϵ)*-HSP oracle of the full-rank lattice* $\Lambda \subset \mathbb{R}^m$*, with* $r \leq \lambda_1(\Lambda)/6$ *and* $\epsilon < 1/4$*. Let* $\mathcal{D}_{\mathbf{f}}$ *be the distribution supported by* Λ^**, with weight* $\langle c_{\ell^*}|c_{\ell^*}\rangle$ *at* $\ell^* \in \Lambda^*$*, where* $|c_{\ell^*}\rangle$ *are the vectorial*

Fourier coefficients of the function $\mathbf{f}$*.*
Then, with overwhelming probability, we need at most

$$O\Big(m \log_2 \left(ma \cdot \det(\Lambda)^{1/m}\right)\Big)$$

samples from $\mathcal{D}_{\mathbf{f}}$ *to fully generate the lattice* Λ^**.*

Proof. Apply Proposition 3.24 with $R = \sqrt{m} \cdot \operatorname{Lip}(\mathbf{f})$ to deduce that $\mathcal{D}_{\mathbf{f}}$ is 3/4-evenly distributed and $(\sqrt{m} \operatorname{Lip}(\mathbf{f}), 1/(4\pi^2))$-concentrated. Subsequently, we apply Lemma 3.21 with[2] $p = 3/4$, $q = 1/(4\pi^2)$, $R = \sqrt{m} \cdot \operatorname{Lip}(\mathbf{f})$ and $t = m \log_2(\sqrt{m} \operatorname{Lip}(\mathbf{f})) - \log_2(\det(\Lambda^*))$, to obtain

$$\Pr\left[S > (2 + \alpha) \cdot 5 \cdot (t + m)\right] \leq \exp(-\alpha(t + m)/2).$$

Writing out t (which is larger than 0), noticing that $\operatorname{Lip}(\mathbf{f}) \leq a$, and absorbing m into the big-O, we obtain the result with exponentially small error probability. □

3.7. Recovering a Basis of the Primal Lattice

The last problem that needs to be resolved is how to obtain an approximate basis $\tilde{B}$ of the primal lattice Λ, given a set of approximate generators $\tilde{G}$ of the dual lattice Λ^*. Also, we would like to know how the approximation errors of $\tilde{G}$ and $\tilde{B}$ are related.

Recovering the approximate basis $\tilde{B}$ proceeds by two steps. The first step consists of applying the Buchmann-Pohst algorithm [BP89] twice to the set of generators $\tilde{G}$, yielding an approximate basis $\tilde{D}$ of the dual lattice Λ^* whose errors are relatively easy to analyze. The second step consists of inverting and transposing the square matrix $\tilde{D}$. This yields an approximate basis $\tilde{B}$ for the primal lattice Λ.

---

[2] In order to apply Lemma 3.21, we need to verify that $R = \sqrt{m} \operatorname{Lip}(\mathbf{f}) \geq \det(\Lambda^*)^{1/m}$. By Remark 3.5, we have $\sqrt{m} \operatorname{Lip}(\mathbf{f}) \geq \frac{\sqrt{m}(1-\epsilon)}{r} \geq \frac{3\sqrt{m}}{\lambda_1(\Lambda)} \geq 3 \det(\Lambda)^{-1/m} = 3 \det(\Lambda^*)^{1/m}$, where we use $r \leq \lambda_1(\Lambda)/6$ and Minkowski's inequality.

The next two subsections follow above summary, and consist of theorems that indicate the decline in precision after each step.

In this particular section, we use row notation for matrices, i.e., any row represents a vector. The matrix of generators $\tilde{G}$ is an $k \times m$ matrix, thus consisting of k generators. We assume that the lattice Λ (and thus Λ^* as well) is of full rank m, meaning that $k \geq m$ and that the resulting bases $\tilde{D}$ and $\tilde{B}$ must be $m \times m$ square matrices. We denote by $\|M\|_\infty$ the matrix norm induced by the infinity norm, explicitly defined as

$$\|M\|_\infty := \max_{1 \leq i \leq m} \sum_{j=1}^{n} |m_{ij}|.$$

3.7.1. An Approximate Well-conditioned Basis of the Dual

Obtaining an approximate and well-conditioned basis of the dual proceeds by means of the Buchmann-Pohst algorithm, which is rigorously analyzed by Buchmann and Kessler [BK96, Sec. 4]. This algorithm consists of concatenating the generating matrix by a scaled identity matrix and applying the LLL lattice reduction algorithm. As described after the proof of [BK96, Thm. 4], this particular algorithm is actually applied twice, once on the matrix of generators $\tilde{G}$ and once again on the resulting intermediate approximate basis $\tilde{D}$ to achieve a new basis whose errors are easier to analyze. From now on, we will refer to applying this procedure twice as the Buchmann-Pohst algorithm. From [BK96] we can extract the following result.

Theorem 3.26. *Let $\tilde{G} = G + \Delta_G$ be an approximation of a $k \times m$ matrix of generators G of the full-rank lattice Λ^* , with $\|\Delta_G\|_\infty < \gamma < \frac{\lambda_1^* \cdot \det(\Lambda^*)}{2^{O(km)} \cdot \|\tilde{G}^*\|_\infty^m}$. Then, on input*[3] *$[\tilde{G} \mid \gamma \cdot I]$, the Buchmann-Pohst algorithm outputs an LLL-reduced matrix $[\tilde{D} | \ \gamma \cdot M]$, with $\tilde{D} = D + \Delta_D$ being an approximate basis of Λ^*, where both $\|\Delta_D\|_\infty$ and $\|\gamma \cdot M\|_\infty$ are upper bounded by*

$$\frac{2^{O(km)} \|\tilde{G}^*\|_\infty^{m+1}}{\lambda_1^* \cdot \det(\Lambda^*)} \cdot \gamma$$

---

[3] Here, the I is the $k \times k$ identity matrix.

Proof. As already mentioned, applying the Buchmann-Pohst algorithm on $\tilde{G}$ takes two reduction steps. The first reduction step is applied on $[\tilde{G}|\ \gamma \cdot I]$ and yields an intermediate basis $\tilde{D}_0 = M_0\tilde{G}$ and the second step is applied on $[\tilde{D}_0 \mid \gamma I]$ and yields the final basis $\tilde{D} = M\tilde{D}_0 = MM_0\tilde{G}$. Here, M and M_0 are unimodular matrices.

The fact that the matrix $[\tilde{D}|\gamma \cdot M]$ is the output of the second step, proves that this must be an LLL-reduced basis (note that $\gamma \cdot M$ is just the matrix M scaled by the scalar γ). From [BK96, Cor. 4.1], we deduce that both $\|M\|_\infty$ and $\|MM_0\|_\infty$ are bounded by $2^{k-1}(\sqrt{mk}+2) \cdot \lambda'\alpha'/\lambda_1(\Lambda^*)$, given that[4] $\gamma < \frac{\lambda_1(\Lambda^*)\det(\Lambda^*)}{(\sqrt{mk}+2)\cdot\lambda'\cdot 2^{\frac{k-3}{2}}}$. Putting in the actual values of $\alpha' = (\sqrt{mk}+2)2^{\frac{k-1}{2}} \cdot \|\tilde{G}\|_\infty$ and

$$\lambda' = \lambda(\sqrt{mk}+2)^m 2^{\frac{k-1}{2}m} = (k\sqrt{m}/2+\sqrt{k})(\sqrt{mk}+2)^m 2^{\frac{k-1}{2}m}\frac{\|\tilde{G}\|_\infty^m}{\det(\Lambda^*)}$$

yields the bound on $\|\gamma M\|_\infty$ and the assumption on γ. For the bound on Δ_D, notice that $\|\Delta_D\|_\infty = \|MM_0\Delta_G\|_\infty \le \|MM_0\|_\infty\|\Delta_G\|_\infty$ and by assumption $\|\Delta_G\|_\infty \le \gamma$. $\square$

For small enough γ, the LLL-reduced basis $[\tilde{D} \mid \gamma \cdot M]$ is very close to $[D \mid 0]$. One of the main results of Chang, Stehlé and Villard [CSV12, Cor. 5.7] states that the close matrix $[D \mid 0]$ must then also be 'weakly LLL-reduced'. This knowledge can then be used to show that this basis D is well-conditioned.

Lemma 3.27. *Let* $[\tilde{D} \mid \gamma M] = [D \mid 0] + [\Delta_D \mid \gamma M]$ *be an LLL-reduced basis with* $\|[\Delta_D \mid \gamma M]\|_\infty \le \mu \cdot (3/\sqrt{2})^{-3m}\|\tilde{D}\|_\infty$ *for some* $\mu < 1$. *Then* D *is* (d, η', θ')*-weakly LLL-reduced as in [CSV12, Def. 5.1], with* $d = \frac{3}{4} + O(2^{-m}\mu)$, $\eta = \frac{1}{2} + O(2^{-m}\mu)$ *and* $\theta = O(2^{-m}\mu)$.

Corollary 3.28. *Let* $[\tilde{D} \mid \gamma M] = [D \mid 0] + [\Delta_D \mid \gamma M]$ *be an LLL-reduced basis with* $\|[\Delta_D \mid \gamma M]\|_\infty \le \mu \cdot (3/\sqrt{2})^{-3m}\|\tilde{D}\|_\infty$ *for some* $\mu < 1$ *(i.e.,*

---

[4] See [BK96, Thm. 4.1], where λ needs to be replaced by λ', as described in the text after the proof of [BK96, Thm. 4.2]). These variables λ and λ' are from [BK96, Prop. 3.2], and not directly related to the minima of the lattices involved.

satisfies the same assumptions as in Lemma 3.27). Then

$$\|D^{-1}\|_\infty \leq \frac{8^m}{\lambda_1(\Lambda^*)}.$$

Proof. We can decompose $D = RVQ$, with Q orthonormal, V diagonal with diagonal entries $\|d_i^*\|$ and R lower triangular with ones on the diagonal. Here, d_i^* are the Gram-Schmidt orthogonalized basis vectors of D.

By the fact that the matrix norm is submultiplicative, we have

$$\|D^{-1}\|_\infty \leq \|R^{-1}\|_\infty \|V^{-1}\|_\infty \|Q^{-1}\|_\infty = \|R^{-1}\|_\infty \|V^{-1}\|_\infty \leq \frac{\|R^{-1}\|_\infty}{\min_i \|d_i^*\|}.$$

By Lemma 3.27, D is weakly (d, η, θ)-LLL-reduced with $d = \frac{3}{4} + O(2^{-m}\mu)$, $\eta = \frac{1}{2} + O(2^{-m}\mu)$ and $\theta = O(2^{-m}\mu)$. Therefore, by [CSV12, Thm. 5.4], taking $\alpha = 2 > \sqrt{2}$ for simplicity, we know that $\lambda_1(\Lambda^*) \leq \|d_1\| \leq 2^m \min_i \|d_i^*\|$, so that $1/\min_i \|d_i^*\| \leq 2^m \lambda_1(\Lambda^*)^{-1}$. In the end of the proof of [CSV12, Lm. 5.5], we see[5] that

$$\|R^{-1}\|_\infty \leq \frac{(1+\alpha)(1+\eta+\theta)^m \alpha^m}{(1+\eta+\theta)\alpha - 1} \leq 4^m,$$

by taking $\alpha = 2, \eta = 1/2 + O(2^{-m}\mu)$ and $\theta = O(2^{-m}\mu)$. This yields the claim. ☐

3.7.2. Inverting the Dual Approximate Basis

As the basis $\tilde{D}$ constructed in the previous subsection is a basis of the *dual* lattice Λ^*, we need to invert and transpose it to get an approximate basis of the primal lattice Λ. In other words, the basis that we would like to approximate is $B = D^{-\top}$, by means of computing $\tilde{B} = \tilde{D}^{-\top}$. Though, inverting an approximate matrix induces errors closely related with the matrix norm of the inverse of the exact basis. More precisely, we have the following result [BKK17, Cor. 7.2, Eq. (7.46)]

---

[5] In [CSV12, Lm. 5.5], the unit-diagonal lower triangular matrix is denoted $\bar{R}$, and the bound is about $\bar{R}^{-1}$

Theorem 3.29. *Let $\tilde{D} = D + \Delta_D$ with $\|\Delta_D\|_\infty \cdot \|D^{-1}\|_\infty < \frac{1}{2}$, and denote $B = D^{-\top}$ and $\tilde{B} = \tilde{D}^{-\top}$ (where $D^{-\top}$ is the inverse transpose of D). Then we have*

$$\|B - \tilde{B}\|_\infty \leq 2\|D^{-1}\|^2 \|\Delta_D\|_\infty.$$

3.7.3. Combining the Errors and Tuning the Parameters

Theorem 3.10. *There exists a polynomial time algorithm, that, for any matrix $G \in \mathbb{R}^{k \times m}$ of k generators of a (dual) lattice Λ^*, and given an approximation $\tilde{G} = G + \Delta_G \in \mathbb{Q}^{k \times n}$, computes an approximation $\tilde{B} = B + \Delta_B$ of a basis B of the primal lattice Λ, such that*

$$\|\Delta_B\|_\infty \leq \frac{2^{O(mk)} \cdot \|\tilde{G}\|_\infty^{m+1}}{(\lambda_1^*)^3 \cdot \det(\Lambda^*)} \cdot \|\Delta_G\|_\infty,$$

under the assumption that $\|\Delta_G\|_\infty < \frac{\min(1, (\lambda_1^*)^2) \cdot \det(\Lambda^*)}{2^{O(km)} \cdot \|\tilde{G}\|_\infty^{m+1}}$.

Proof. For the moment, assume that the full output[6] $[\tilde{D} \mid \gamma M] = [D \mid 0] + [\Delta_D \mid \gamma M]$ of the Buchmann-Pohst algorithm satisfies $\|[\Delta_D \mid \gamma M]\|_\infty \leq \mu(3/\sqrt{2})^{-3m}\|\tilde{D}\|_\infty$ for some $\mu < 1$ and $\|\Delta_D\|_\infty \|D^{-1}\|_\infty < 1/2$. Then, by applying Theorem 3.29, Corollary 3.28 and Theorem 3.26 subsequently, we obtain

$$\|\Delta_B\|_\infty \leq 2\|D^{-1}\|_\infty^2 \|\Delta_D\|_\infty \leq \frac{2^{6m+1}}{(\lambda_1^*)^2}\|\Delta_D\|_\infty \leq \frac{2^{O(mk)} \cdot \|\tilde{G}\|_\infty^{m+1}}{(\lambda_1^*)^3 \cdot \det(\Lambda^*)} \cdot \gamma.$$

It remains to prove that assumptions in the beginning of this proof are indeed fulfilled. By Theorem 3.26, we have

$$\|[\Delta_D \mid \gamma M]\|_\infty \leq \frac{2^{O(mk)} \cdot \|\tilde{G}\|_\infty^{m+1}}{\lambda_1^* \cdot \det(\Lambda^*)} \cdot \gamma < O(1),$$

---

[6] In reality, the Buchmann-Pohst algorithm is applied with the largest precision such that all required assumptions hold. So the costs of applying the LLL-algorithm does not involve the precision $\|\Delta_G\|_\infty$.

and by Theorem 3.29, we have

$$\|\Delta_D\|_\infty \|D^{-1}\|_\infty \leq \|\Delta_D\|_\infty \frac{2^{3m}}{\lambda_1^*} \leq \frac{2^{O(mk)} \cdot \|\tilde{G}\|_\infty^{m+1}}{(\lambda_1^*)^2 \cdot \det(\Lambda^*)} \cdot \gamma < O(1).$$

So choosing γ appropriately small, the assumptions of Theorem 3.29, Corollary 3.28 and Theorem 3.26 are all fulfilled. □