Conference Paper

Sampling algorithms for lattice Gaussian codes

Author[s]:
Campello, Antonio; Belfiore, Jean-Claude

Publication Date:
2016

Permanent Link:
https://doi.org/10.3929/ethz-a-010645959

Rights / License:
In Copyright - Non-Commercial Use Permitted
Sampling Algorithms for Lattice Gaussian Codes

Antonio Campello
and Jean-Claude Belfiore
Communications & Electronics Department
Télécom ParisTech
Paris, France
Email: {campello,belfiore}@telecom-paristech.fr

Abstract—We consider the problem of sampling a discrete Gaussian distribution whose support is an $n$-dimensional lattice. Fast sampling algorithms for lattices decomposed as a finite union of cosets are proposed. This includes the low dimensional lattices with the best coding gains, their duals, and the 24 dimensional Leech lattice. Our methods are then applied to assess the performance of recent sampling-based codes for the AWGN channel, illustrating the gains of the discrete Gaussian distribution.

In the derivation of our algorithms, a number of results concerning the theta series of notable lattices will be discussed, including relations between the theta series and its derivatives to the power and rate of a lattice Gaussian code.

I. INTRODUCTION

In recent capacity achieving lattice coding schemes for the AWGN channel [7] and for the semantically secure wiretap channel [8], the sent vector is drawn from a discrete Gaussian distribution whose support is a multidimensional Euclidean lattice, as in Figure 1. Other applications of such a distribution include some of the state-of-the-art lattice-based cryptographic models (e.g. [11], [2]), and the generation of information theoretic secure secret keys [9].

A. Lattices

We consider real and complex lattices. A real lattice $\Lambda$ is a discrete additive subgroup of $\mathbb{R}^n$, whereas a complex lattice is a discrete additive subgroup of $\mathbb{C}^n$. A complex lattice can be always identified with a real lattice in $\mathbb{R}^{2n}$ in a straightforward way by considering the real and imaginary parts. For example, when $\omega = -1/2 + \sqrt{3}/2i$, the lattice of Eisenstein integers $\mathbb{Z}[\omega] = \{a + b\omega : a, b \in \mathbb{Z}\}$ is identified with the real hexagonal plane lattice denote by $A_2$ [4].

There are classical ways of constructing lattices from error correcting codes. Let $C \subset \mathbb{F}_2^n$ be a linear code and $P \subset \mathbb{F}_2^n$ be the parity-check code. By identifying the elements of $\mathbb{F}_2^n$ with their binary expansions, the cosets of the code $C$ can be identified with the cosets of the lattice $P$. This is not a trivial task, and even unidimensional samplers for the lattice Gaussian distribution have been object of research (see [5], [6] and [2, Sec. 5.1]) - in fact, most multidimensional samplers use them as sub-routines. An obstacle for these algorithms is sampling over Gaussians which are not sufficiently flat, i.e., when the variance parameter $\sigma$ is small or moderate.

In this work we focus on specialized algorithms for lattices commonly used for coding. We propose algorithms for sampling on lattices obtained from constructions A, B, their complex versions, and the density doubling construction. This includes the low dimensional lattices with best coding gain, their duals, and the 24 dimensional Leech lattice. In the derivation of our algorithms, a number of results concerning the theta series of these lattices and their relations to coding parameters will be discussed. Particularly interesting are closed form expressions for the power and rate of a lattice Gaussian code (Prop. 5) in terms of the theta series and its derivative.

Our algorithms output the correct distribution for the specific lattices and any $\sigma$, comparing favorably to universal Markov Chain based algorithms like [14]. For a concrete example, sampling within statistical distance $10^{-3}$ from the centered discrete Gaussian over the Leech lattice $A_{24}$ in the worst case $\sigma = 1/\sqrt{2\pi}$ requires 13434 iterations (cf. [14, Eq. 26 and Lem. 3]), or $24 \times 13434 = 322416$ calls of an unidimensional $Z$-sampler. In a huge contrast, the number of calls of the $Z$-sampler for our tailor-made Leech lattice sampler is 24. This is the same contrast between universal decoders (e.g. the sphere decoder) and specialized decoders for particular lattices (e.g., root lattices, etc.).

The rest of this paper is organized as follows. In Section III we review unidimensional samplers for cosets of the lattice $\mathbb{Z} + c$. In Section IV, we describe a general principle for lattices decomposed as the union of cosets, which is then applied in Sections V-VII to several lattices obtained from codes. In Section VIII, we apply our algorithm to assessing the codeword error probability performance of lattice Gaussian codes [7] for a code based on the Leech lattice $A_{24}$.

Fig. 1. Distribution obtained from the $A_2$ sampler (Sec V). Blue and red dots correspond to points in $\mathbb{Z} \oplus \sqrt{3}\mathbb{Z}$ and $\mathbb{Z} \oplus \sqrt{3}\mathbb{Z} + (1/2, \sqrt{3}/2)$, respectively.

II. PRELIMINARIES AND NOTATION

A. Lattices

We consider real and complex lattices. A real lattice $\Lambda$ is a discrete additive subgroup of $\mathbb{R}^n$, whereas a complex lattice is a discrete additive subgroup of $\mathbb{C}^n$. A complex lattice can be always identified with a real lattice in $\mathbb{R}^{2n}$ in a straightforward way by considering the real and imaginary parts. For example, when $\omega = -1/2 + \sqrt{3}/2i$, the lattice of Eisenstein integers $\mathbb{Z}[\omega] = \{a + b\omega : a, b \in \mathbb{Z}\}$ is identified with the real hexagonal plane lattice denote by $A_2$ [4].

There are classical ways of constructing lattices from error correcting codes. Let $C \subset \mathbb{F}_2^n$ be a linear code and $P \subset \mathbb{F}_2^n$ be the parity-check code. By identifying the elements of $\mathbb{F}_2^n$ with their binary expansions, the cosets of the code $C$ can be identified with the cosets of the lattice $P$. This is not a trivial task, and even unidimensional samplers for the lattice Gaussian distribution have been object of research (see [5], [6] and [2, Sec. 5.1]) - in fact, most multidimensional samplers use them as sub-routines. An obstacle for these algorithms is sampling over Gaussians which are not sufficiently flat, i.e., when the variance parameter $\sigma$ is small or moderate.

In this work we focus on specialized algorithms for lattices commonly used for coding. We propose algorithms for sampling on lattices obtained from constructions A, B, their complex versions, and the density doubling construction. This includes the low dimensional lattices with best coding gain, their duals, and the 24 dimensional Leech lattice. In the derivation of our algorithms, a number of results concerning the theta series of these lattices and their relations to coding parameters will be discussed. Particularly interesting are closed form expressions for the power and rate of a lattice Gaussian code (Prop. 5) in terms of the theta series and its derivative.

Our algorithms output the correct distribution for the specific lattices and any $\sigma$, comparing favorably to universal Markov Chain based algorithms like [14]. For a concrete example, sampling within statistical distance $10^{-3}$ from the centered discrete Gaussian over the Leech lattice $A_{24}$ in the worst case $\sigma = 1/\sqrt{2\pi}$ requires 13434 iterations (cf. [14, Eq. 26 and Lem. 3]), or $24 \times 13434 = 322416$ calls of an unidimensional $Z$-sampler. In a huge contrast, the number of calls of the $Z$-sampler for our tailor-made Leech lattice sampler is 24. This is the same contrast between universal decoders (e.g. the sphere decoder) and specialized decoders for particular lattices (e.g., root lattices, etc.).

The rest of this paper is organized as follows. In Section III we review unidimensional samplers for cosets of the lattice $\mathbb{Z} + c$. In Section IV, we describe a general principle for lattices decomposed as the union of cosets, which is then applied in Sections V-VII to several lattices obtained from codes. In Section VIII, we apply our algorithm to assessing the codeword error probability performance of lattice Gaussian codes [7] for a code based on the Leech lattice $A_{24}$.
as elements of \( \mathbb{Z} \) we can write the constructions A and B of \( C \) via the “code-formulas”:

\[
\Lambda_A(C) = 2Z^n + C \quad \text{and} \quad \Lambda_B(C) = 4Z^n + 2P_n + C. \quad (1)
\]

Alternatively \( \Lambda_B(C) = 2D_n + C \), where \( D_n = \Lambda_A(P_n) \). We can write analogous formulas for complex lattices:

\[
\Lambda_A(C) = \theta Z[\omega]^n + C \quad \text{and} \quad \Lambda_B(C) = \theta^2 Z[\omega]^n + \theta P^n + C, \quad (2)
\]

where \( \theta \in \mathbb{Z}[\omega] \) is a prime with norm \( |\theta|^2 = p \) and \( C \subset \mathbb{Z}_p \) is a linear code. Again, \( \Lambda_B(C) = \theta \Lambda_A + C \), where \( \Lambda_A = \Lambda_A(P_n) \).

### B. Jacobi Theta Functions

Let \( \Lambda \) be a lattice and \( \mathbf{e} \) a vector in \( \mathbb{R}^n \). The theta series of \( \Lambda + \mathbf{e} \) is defined as:

\[
\Theta_{\Lambda+\mathbf{e}}(\tau) := \sum_{\mathbf{y} \in \Lambda+\mathbf{e}} e^{-\pi \tau |\mathbf{y}|^2} = \sum_{\mathbf{x} \in \Lambda} e^{-2\pi \tau \langle \mathbf{x}+\mathbf{e}, \mathbf{x}+\mathbf{e} \rangle}.
\]

The theta series provides useful information about a lattice, such as its minimal norm, kissing number and determinant (for undefined terms, see [4]). In Communications, the theta series bounds the probability of error of a lattice code used for the Gaussian channel (see, e.g., [4, Ch. 3, Eq. (35)]) and for the Gaussian wiretap channel [10] among other applications.

The theta series of all lattices discussed in this paper can be written in terms of the Jacobi theta functions (in what follows, let \( q = e^{-\pi \tau}, \tau > 0 \) and \( z = i\tau \)):

\[
\theta_3(z) := \sum_{m=-\infty}^{\infty} e^{2\pi im^2 z} = \sum_{m=-\infty}^{\infty} \cos(2\pi mz) e^{\pi i mz^2}, \quad \theta_2(z) := \sum_{m=-\infty}^{\infty} q^{m+1/2}.
\]

Notice that \( \theta_2(z) \) and \( \theta_3(z) \) are the theta series of the unidimensional lattice \( \mathbb{Z} \) and its shift \( \mathbb{Z} + 1/2 \), respectively. More generally (see Eq. (2.2.5) of [1]):

\[
\Theta_{\mathbb{Z}+\mathbf{c}}(\tau) = \sum_{m=-\infty}^{\infty} e^{-\pi \tau (m+\mathbf{c})^2} = \tau^{-2} \sum_{m=-\infty}^{\infty} e^{2\pi i mc - \pi m^2/\tau} = \tau^{-2} \theta_3(\tau c) |\tau|^{-1}.
\]

For numerical aspects and efficient evaluations of the Jacobi series the reader is referred to [1, Ch. 2-3].

### C. Discrete Gaussian Distributions

Define the Gaussian function \( \rho_{\sigma}(x) = e^{-|x|^2/2\sigma^2} \) and, for a discrete set \( S \subset \mathbb{R}^n \), let

\[
\rho_{\sigma}(S) := \sum_{x \in S} \rho_{\sigma}(x).
\]

The **discrete Gaussian distribution** over \( \Lambda + \mathbf{c} \) is defined as the distribution with support in \( \Lambda + \mathbf{c} \), such that the probability of choosing a vector \( y \in \Lambda + \mathbf{c} \) is proportional to \( \rho_{\sigma}(y) \). We denote the probability that a random vector drawn according to the discrete Gaussian distribution is equal to \( y \in \Lambda + \mathbf{c} \) by

\[
D_{\Lambda + \mathbf{c}, \sigma}(y) := \frac{\rho_{\sigma}(y)}{\rho_{\sigma}(\Lambda + \mathbf{c})} = \frac{\rho_{\sigma}(x + \mathbf{c})}{\theta_{\Lambda+\mathbf{c}}(\tau c)}
\]

Some simple but useful properties of \( D_{\Lambda + \mathbf{e}, \sigma}(y) \) are stated next:

**Proposition 1.** The lattice Gaussian distribution satisfies:

(i) \( D_{\Lambda+\mathbf{c}, \sigma}(\mathbf{y}) = D_{\Lambda+\mathbf{e}, \sigma}(\mathbf{y}) \).

(ii) \( D_{\Lambda+\mathbf{c}}, \| (\Lambda+\mathbf{e}) (\mathbf{y}_1, \mathbf{y}_2) = D_{\Lambda+\mathbf{c}}(\mathbf{y}_1) D_{\Lambda+\mathbf{e}}(\mathbf{y}_2) \).

A lattice Gaussian sampler is an algorithm that outputs a point \( y \in \Lambda + \mathbf{c} \) with probability \( D_{\Lambda+\mathbf{e}, \sigma}(y) \).

### III. Building Blocks: Gaussians over \( \mathbb{Z} + \mathbf{c} \)

Unidimensional discrete Gaussians are the building blocks for the main multi-dimensional samplers, including the ones described in this paper. Efficient practical samplers over \( \mathbb{Z} \) can be found e.g., in [5], [6] and all these methods can be used as subroutines our algorithms. A theoretical method described in [2] shows that it is possible to output the exact distribution \( D_{\mathbb{Z}+\mathbf{c}} \) by calling a continuous Gaussian sampler and using a rejection principle. The expected number of iterations is \( [2, \text{Sec. 5.1}] \) (consider \( 0 < c < 1 \) for simplicity):

\[
\rho_{\sigma}(c) + \rho_{\sigma}(1-c) + \int_1^\infty \rho_{\sigma}(x) dx + \int_{-\infty}^{1-c} \rho_{\sigma}(x) dx.
\]

Using Equation (4) we can prove that the expected number of iterations tend to 1 as \( \sigma \rightarrow 0 \) or \( \sigma \rightarrow 1 \). Numerically evaluations for the probability of acceptance (inverse of expected iterations) are shown in Fig. 2.

![Fig. 2. Acceptance probability as a function of \( \sigma \) for fixed \( l = 1 \) (left) and fixed \( c = 0 \) (right). In the worst case, \( c = 0 \) and \( \sigma \approx 0.12680 \), the average number of iterations is no bigger than 1.426764](image)

### IV. Coset Decompositions

Suppose that the lattice \( \Lambda \) can be decomposed as the disjoint union of cosets \( \Lambda = \bigcup_{x \in \Lambda'} \Lambda' + x \). Let \( p_c = D_{\Lambda, \sigma}(c + \Lambda') \) be the probability that a point drawn from a discrete Gaussian in \( \Lambda \) lies in the coset \( \Lambda' + x \). A general principle for sampling \( \Lambda \) is the following:

1. Pick a vector \( c \) at random, with probability \( p_c \).
2. Pick a vector from \( D_{\Lambda'+c, \sigma} \) and output it.

The procedure outputs a point \( c + x \in c + \Lambda' \) with the correct probability \( p_c D_{\Lambda'+c, \sigma}(c + x) \). To apply the general principle we to calculate the probabilities \( p_c \), samplers for shifts of the superlattice \( \Lambda' \) and a systematic description of the cosets.

The following table is a collection of results on the theta series of some lattices constructed from codes. To facilitate the statements, let

\[
\phi_0(\tau) := \theta_3(\tau) \theta_3(3\tau) + \theta_2(\tau) \theta_2(3\tau),
\]
\( \phi_1(\tau) := \theta_2(\tau)\theta_3(3\tau) + \theta_3(\tau)\theta_2(3\tau) = \frac{1}{2} \theta_2(\tau/4)\theta_2(3\tau/4) \), and rectangular lattice \( \mathbb{Z} \oplus \sqrt{3}\mathbb{Z} \) is a sublattice of \( A_2 \) of index 2. We can write

\[
A_2 = \left( \mathbb{Z} \oplus \sqrt{3}\mathbb{Z} \right) \bigcup \left( \mathbb{Z} \oplus \sqrt{3}\mathbb{Z} + \left( \frac{1}{2}, \frac{\sqrt{3}}{2} \right) \right)
\]

From this:

\[
D_{A_{2,\sigma}}(\mathbb{Z} \oplus \sqrt{3}\mathbb{Z}) = \frac{\theta_3(\frac{1}{2\pi \sigma})\theta_3(\frac{3}{2\pi \sigma})}{\theta_3(\frac{1}{2\pi \sigma})\theta_3(\frac{3}{2\pi \sigma}) + \theta_3(\frac{1}{2\pi \sigma})\theta_3(\frac{1}{2\pi \sigma})}.
\]

Note that if \( \sigma \) is small, all mass is concentrated in the origin, hence \( D_{A_{2,\sigma}}(\mathbb{Z} \oplus \sqrt{3}\mathbb{Z}) \approx 1 \), and if \( \sigma \) is large, the distribution is flat, therefore \( D_{A_{2,\sigma}}(\mathbb{Z} \oplus \sqrt{3}\mathbb{Z}) \approx 1/2 \).

\( B. \) The lattice \( D_n \) and the shift \( D_n + (\alpha, \beta, \ldots, \beta) \)

Using Construction A, write \( D_n = 2\mathbb{Z}^n + \mathbb{C} \), where \( \mathbb{C} \) is a parity-check code \( (n, n - 1, 2) \). For sampling Construction B we need samplers over a shift of \( D_n \) lattice, by a vector of the form \( (\alpha, \beta, \ldots, \beta) \). We begin by calculating the theta series of the shift via Construction A. Some of these calculations can be found in [4] for \( (\alpha, \beta) = (0, 0), (1/2, 1, 2), \) and \( (1, 0) \).

**Proposition 3.** Let

\[
W_n(X, Y) = \sum_{i=1}^{\lfloor (n-1)/2 \rfloor} \left( \frac{n-1}{2l} \right) Y^{2l} X^{n-2l} \quad \text{and}
\]

\[
W_n(X, Y) = \sum_{i=1}^{\lfloor (n+1)/2 \rfloor} \left( \frac{n+1}{2l-1} \right) Y^{2l-1} X^{n-2l}
\]

be the weight enumerators of the vectors in \( \mathbb{F}_2^n \) with even and odd weights, respectively. We have

\[
\Theta_{D_n + (\alpha, \beta, \ldots, \beta)}(q) = \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^4) W_n(q) \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^4) + \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^{4}) W_n(q^2) \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^{4}).
\]

The case \( \beta = 1/2 \) is very particular, since the theta series of all cosets \( c + (1/2, 2, \ldots, 1/2 + 2\mathbb{Z}^n) \) are equal. Hence, a simple sampler in this case is obtained by sampling a word \( c \) uniformly at random over all codewords in \( \mathbb{C} \) and then sampling over the lattice \( c + (1/2, 2, \ldots, 1/2 + 2\mathbb{Z}^n) \). This procedure was described in [7]. For \( \alpha = 1/2 \), let

\[
p_{\text{tw}} := \left( \frac{n}{2l} \right) \frac{\Theta_{\mathbb{Z}^n + \mathbb{C}}(q^{4} \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^{4}) \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^{4}))}{\Theta_{D_n,\beta}(q)}
\]

be the probability that a vector drawn from distribution \( D_{D_n,\beta,\sigma} \) lies in the coset of a codeword of weight \( w \). Algorithm 7 provides a sampling procedure for \( D_{D_n,\beta,\sigma} \). In the algorithm let \( \mathbb{I}_2 = \{1, \ldots, n\} \).

The case \( \beta \neq 1/2 \) is very similar to the previous one, except that we have to distinguish codewords with \( c_1 = 0 \) and \( c_1 = 1 \). Given that a point lies in a coset of a codeword of weight \( w \), the probability that this codeword has \( c_1 = 0 \) is given by:

\[
p_c := \frac{\Theta_{\mathbb{Z}^n + \mathbb{C}}(q^4) \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^4) + \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^{4}) \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^{4})}{\Theta_{\mathbb{Z}^n + \mathbb{C}}(q^4) \Theta_{\mathbb{Z}^n + \mathbb{C}}(q^4)}
\]
Sampler\(\text{DN}(\beta, \sigma)\)
1: Choose a number \(l\) from 1 to \(\lfloor n/2 \rfloor\) with probability \(p_{2l}\).
2: Choose uniformly at random a set \(J \subset I_n\), with size 2\(l\).
3: For \(j \in J\),
4: \(x_j \leftarrow 2\text{Sampler}\(Z((\beta + 1)/2, \sigma/2)\)\).
5: For \(j \in I_n \setminus J\),
6: \(x_j \leftarrow 2\text{Sampler}\(\beta, \sigma/2)\).
7: Output: \(x = (x_1, x_2, \ldots, x_n)\).

Thus a modification of Algorithm 7 can be done as follows. After picking a weight 2\(l\) according to the right probability, we throw a biased coin with probability \(p_{2l}\) of heads. If the result is heads, then choose \(c_1 = 0\), choose a set \(J \subset I_{n-1}\) with \(|J| = 2\lfloor l/2 \rfloor\) to be the support of \((c_2, \ldots, c_n)\) and sample over \(2^{\mathbb{Z}^n}\) to be the support of \((c_2, \ldots, c_n)\) and sample over \(2^\mathbb{Z}^n + c\) as in steps 3-6 of Algorithm 7. If the result is tails, set \(c_1 = 1\) and choose \(J \subset I_{n-1}\), with \(|J| = 2\lfloor l/2 \rfloor - 1\).

VI. CONSTRUCTION B

As in Construction A, the probability of a coset \(2D_n + c\) only depends on the Hamming weight of \(c\), since any permutation of coordinates is an automorphism of \(D_n\). Let \(\mathcal{D}_n = D_n + (1, 0^{n-1})\). To calculate the probability of a coset in a suitable way to sampling, consider the decomposition, for \(1 \leq w \leq n\):

\[
D_n = (D_w + D_{n-w}) \cup (\mathcal{D}_w + \mathcal{D}_{n-w}),
\]

which we refer to as the even-even/odd-odd decomposition. The theta series of \(2(D_n + c/2)\) is

\[
\Theta_{D_n + c/2}(q) = \Theta_{D_n + (1, 0^{n-1})}(q) \Theta_{D_n - (1, 0^{n-1})}(q).
\]

But all the terms in the rhs of (9) are in the form of Section V-B. Let \(p_w = a'_w \Theta_{D_n + (1, 0^{n-w-1})}(q) \Theta_{\lambda}(q)\) and define \(p_{\text{even}, w} = D_{2D_n + c}(2(D_w + 1^w) + \mathcal{D}_n) + \mathcal{D}_n\).

A sampling procedure for Construction B lattices works as follows. Draw a weight \(w\) according to the probabilities \(p_{w}\). Draw a codeword \(c\) uniformly at random over all codewords with weight \(w\) in \(C\). Throw a biased coin with probability \(p_{\text{even}, w}\) of heads, and if the result is heads sample over the even-even part of decomposition (9) (formally, if \(J\) be the support of \(c\), draw \(x_J\) as from the distribution over \(2D_w + (1^w)\) and \(x_{I \setminus J}\) from the distribution over \(2D_{n-w}\)). Otherwise, sample over the odd-odd part of \(D_n\).

VII. THE LEECH LATTICE

We provide a sampling algorithm based on the density doubling construction. Let \(G_{24}\) be the \((24, 12, 8)/2\) Golay code. We write half of the Leech lattice as \(H_{24} = 2D_{24} + G_{24}\). Let \(\alpha = (-3/2)^1, (1/2)^{23}\). The (scaled) Leech lattice is:

\[
\Lambda_{24} = H_{24} \cup (H_{24} + \alpha).
\]

The weight enumerator of \(G_{24}\) is given by

\[
W(X, Y) = X^{24} + 759X^{16}Y^8 + 2576X^{12}Y^{12} + 759X^8Y^{16} + Y^{24}.
\]

The first half of the Leech is a construction \(B\), hence its theta series is

\[
\Theta_{H_{24}}(q) = (1/2)W(q^4)\Theta_4(q^4) + (1/2)\Theta_8(q^8).
\]

and a sampler is provided by the techniques in Section V-B. For the other half, an application of Prop. 3 gives us closed forms. A simpler closed form can be derived from the auxiliary series:

\[
\alpha(q) = \sum_{m=-\infty}^{\infty} (-1)^m q^{(m+1/4)^2} \quad \text{and}
\]

\[
\beta(q) = \Theta_{Z_{24}/4}(q) = \theta_3(q^{1/16}) - \theta_3(q/4) - \theta_3(q/2).
\]

**Proposition 4.** All cosets \(2D_{24} + c + a\) have the same theta series, given by:

\[
\Theta_{2D_{24} + c + a}(q) = \beta(q) - \alpha(q) - \alpha(q)^2.
\]

The other half of the Leech has theta series

\[
2^{11}\beta(q) - 2 \alpha(q) = 98304q^8 + 8388608q^{12} + \ldots.
\]

**Proof.** First note that any permutation of coordinates is an automorphism of \(2D_{24} + a\). Thus, by permuting coordinates, we obtain a set isometric to \(2D_{24} + (1^w, 0^{24-w}) + a \approx 2D_{24} + a\). For calculating the theta series of \(2D_{24} + a\), notice that \(\alpha(q)^n\) takes negative sign in the terms associated to the vectors \(x + (1/4^{24}) \in \mathbb{Z}_{24} + (1/4^{24})\), where \(x\) has odd weight. Hence \(\Theta_{2D_{24} + (1/4^{24})}(q) = (\beta(q) + \alpha(q)^2)/2\). This, together with the equality

\[
(D_{24} + (-3/4)^{24}) \cup (D_{24} + (1/4^{24})) = \mathbb{Z}^{24} + a/2,
\]

gives us the desired form. □

Let \(D_{\Lambda_{24}, \sigma}(H_{24}) = \Theta_{H_{24}}(q)/\Theta_{\Lambda_{24}}(q)\) be the probability that a point sampled from the distribution in \(\Lambda_{24}\) lies in \(H_{24}\). The following procedure outputs a point \(x \in \Lambda_{24}\) distributed according to \(D_{\Lambda_{24}, \sigma}\).

**Sampler \(\Lambda_{24}\)**
1. Throw a biased coin with prob. \(D_{\Lambda_{24}, \sigma}(H_{24})\) of heads.
2. If the output is heads then
3. Sample \(x \in H_{24}\) from the Construction B sampler
4. Else
5. Choose \(c \in G_{24}\) uniformly at random
6. Draw \(x \in 2D_{24} + \alpha + c\) using sampler in Sec. V-B.
7. End if
8. Output \(x\).

Step 3 involves sampling the codewords of \(C\) according to its weight and to probabilities \(p_w\) (as in Sec. IV item 1’). In what follows we provide an alternative and more efficient procedure than listing all the \(2^{24}\) codewords.

First, pick a number \(w \in \{0, 8, 12, 16, 24\}\) according to probabilities \(p_w\). For \(w = 0\) and 24 there is nothing to do. For \(w = 12\), a simple rejection algorithm works, since
2576/1096 = 62% of the codewords in \(G_{24}\) have weight 12. Thus a rejection algorithm performs an average of \(\approx 1.590002\) iterations. However, the same procedure applied to \(w = 8\) yields over 5 iterations with variance 23.813002. A “non-rejection” method uses the well-known fact that minimum weight codewords form a \(S(5,8,24)\) Steiner system [13]. This means that every word of weight 5 in \(G_{24}\) decodes to one and only one codeword of weight 8 in \(G_{24}\). But generating a word of weight 5 in \(G_{24}\) is a simple “k-out-of-n” procedure (pick a subset of size 5 in \(\{1, \ldots, 24\}\), and \(G_{24}\) can be decoded very efficiently. For generating codewords of weight \(w = 16\), just generate a codeword of weight 8 and add the vector \((1, \ldots, 1) \in G_{24}\).

**VIII. The Lattice Gaussian Coding Scheme**

In the scheme of [7], a lattice point \(\mathbf{z}\), chosen according to the distribution \(D_{\Lambda, \mathbf{c}, \sigma}\), is sent over a Gaussian channel. It is proven that if \(\Lambda\) is an AWGN-good lattice and the dimension \(n \to \infty\), transmission rates up to the capacity of the channel can be achieved, provided that \(\text{snr} > c\). The transmission rate and the power are the entropy and variance per dimension of \(D_{\Lambda, \mathbf{c}, \sigma}\), and asymptotic formulas for these quantities are given in [8, Lem. 6-7]. The following proposition shows closed forms based on the theta series of \(\Lambda\) and its derivative.

**Proposition 5.** The rate (in nats per channel use) and power of a Lattice Gaussian Code are given by

\[
P = \frac{-1}{n\pi} \frac{\Theta_{\lambda + e}(\tau)}{\Theta_{\lambda + e}(\tau)} \quad \text{and} \quad R = \frac{\tau}{n} \frac{\Theta_{\lambda + e}(\tau)}{\Theta_{\lambda + e}(\tau)} + \frac{1}{n} \ln \Theta_{\lambda + e}(\tau),
\]

(14)

where \(\tau = 1/2\pi\sigma^2\).

Alternatively, we can write

\[
R = \frac{P}{2\sigma^2} + \ln(\Theta_{\lambda + e}(\tau)^{1/n})
\]

(15)

Notice that if we scale the lattice \(\Lambda\) and \(\sigma\) by \(\alpha > 0\) we increase the power and keep the rate constant. A corollary of Prop. 5 is that the rate is maximized in the centered distribution, i.e., when \(c = 0\).

As an application of the Leech Sampler in Sec VII, we simulate the probability of error of a lattice Gaussian code scheme, in comparison with a cubic shaped Leech codebook. The choice \(\sigma = 0.936797\) in this case leads to \(R = 1.5\) bits (or 1,039,720 nats) and \(P = 0.936797\). Simulations were based in 10^9 samples. Notice that the maximum possible shaping gain of a Voronoi codebook is \(\approx 1.53\) dB, when the dimension \(n \to \infty\).

**ACKNOWLEDGMENT**

The work of AC was supported by FAPESP under grant 2014/20602-8. The authors would like to thank Patrick Solé for helpful discussions on theta series.

**REFERENCES**


