Lattice Coding for Signals and Networks

A Structured Coding Approach to Quantization, Modulation and Multi-user Information Theory

RAM ZAMIR
Tel Aviv University

with contributions by
BOBAK NAZER
Boston University

and
YUVAL KOCHMAN
The Hebrew University of Jerusalem

and with illustrations by
Ilai Bistritz

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9780521766982

First published 2014

Printed in the United Kingdom by Clays, St Ives plc

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data
Zamir, Ram.
Lattice coding for signals and networks : a structured coding approach to quantization, modulation, and multiuser information theory / Ram Zamir, Tel Aviv University.
pages cm
Includes bibliographical references and index.
ISBN 978-0-521-76698-2 (hardback)
1. Coding theory. 2. Signal processing – Mathematics. 3. Lattice theory. I. Title.
TK5102.92.Z357 2014
003′.54 – dc23 2014006008

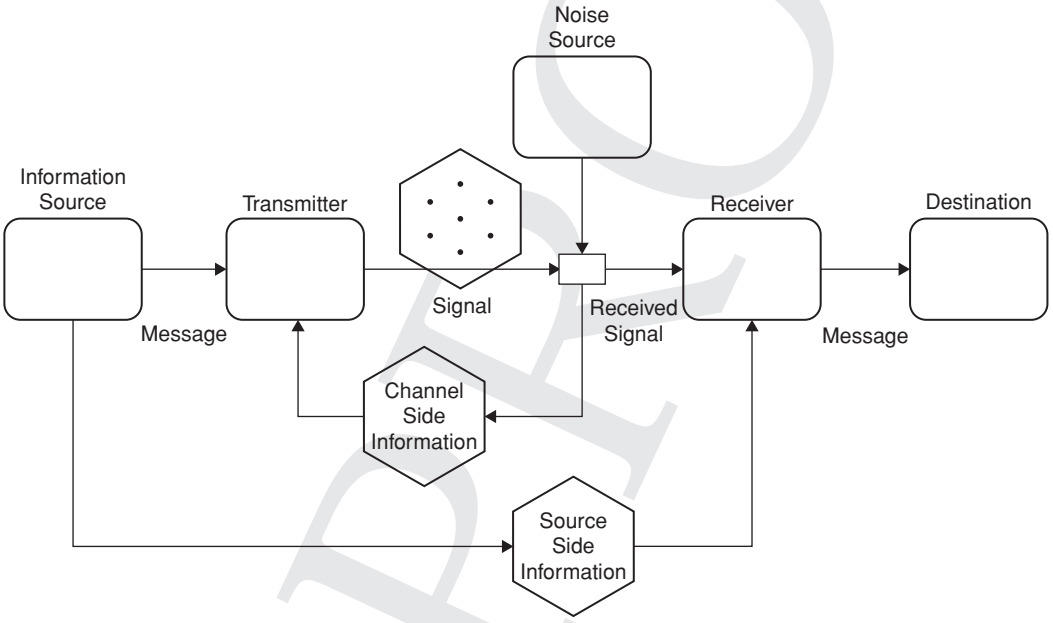
ISBN 978-0-521-76698-2 Hardback

To my parents Eti and Sasson Zamir

Contents

Preface

Digital communication and information theory talk about the same problem from very different aspects. Lattice codes provide a framework to tell their mutual story. They suggest a common view of source and channel coding, and new tools for the analysis of information network problems.



This book makes the language of quantization and modulation more accessible to the hard core information theorist. For him or her, lattices serve as a bridge from the high dimension of Shannon's theory to that of digital communication techniques. At the same time, lattices provide a useful tool for the communication engineer, whose scope is usually limited to the low – sometimes even one or two – dimensions of practical modulation schemes (e.g., QAM or PCM). She or he can "see," through the lattice framework, how signals and noise interact as the dimension increases, for example, when modulation is combined with coding. Surprisingly for both disciplines,

the generalization of the lattice framework to "Gaussian networks" is not only very natural, but in some cases is more powerful than the traditional techniques.

This book is beneficial to the "Gaussian-oriented" information theorist, who wishes to become familiar with network information theory from a constructive viewpoint (as opposed to the more abstract random-coding/random-binning approach). And it is a useful tool for the communication practitioner in the industry, who prefers a "geometric" and "signal-processing oriented" viewpoint of information theory in general, and multi-user problems in particular. The algebraic coding theorist can celebrate the variety of new applications for lattice codes found in the book. The control theorist, who wishes to add communication constraints into the system, will find the linear-additive model of dithered lattice quantization useful. Other readers, like those having a background in signal processing or computer networks, can find potential challenges in the relations to linear estimation and network coding.

Ram Zamir
Tel Aviv
March 2014

Acknowledgements

In the spring of 1989 I took a data compression course with Meir Feder at Tel Aviv University, and read Jacob Ziv's 1985 article "On universal quantization." Ziv showed that the redundancy of a randomized (dithered) uniform scalar quantizer is always bounded by $\approx$0.754 bit; he also stated, without a proof, that if the scalar quantizer is replaced by a "good" high-dimensional lattice quantizer, then this universal bound can be reduced to half a bit – provided that Gersho's conjecture is true. In my final project – while learning about Gersho's conjecture and verifying Ziv's statement – I fell in love with the world of lattice codes.

Many people with whom I have collaborated since then have contributed to the material presented in this book. The roots of the book are in my MSc and PhD research – under Meir's supervision – about entropy-coded dithered lattice quantization (ECDQ) of signals. Tamas Linder offered rigorous proofs for some of the more technical results (and became my colleague and coauthor for many years). Two other staff members in the EE department in Tel Aviv University – Gregory Poltyrev and Simon Litsyn – while contributing from their wide knowledge about lattices, opened the way to the later applications of dithered lattice codes to signaling over the AWGN channel.

Toby Berger – my post-doctoral mentor at Cornell University in the years 1994–1995 – introduced me to the fascinating world of multi-terminal source coding. This became the first instance where randomized lattice codes were applied in network information theory. A year later, Shlomo Shamai and Sergio Verdú, with whom I communicated about systematic lossy source-channel codes, inspired me to introduce the idea of nested lattices for the Wyner–Ziv (source coding with side information) problem. This idea, which started as a toy example for a more practical systematic source-channel code, grew later into a general framework for "algebraic binning" for information networks.

Uri Erez took my first advanced information theory course in the spring of 1997; in his final project he developed an interesting technique for using channel-state information at the transmitter. His PhD research then became a fusion center of many ideas in lattice coding for noisy channels: following a pointer given to us by Shlomo Shamai to the Costa problem, Uri came up with the innovative idea of lattice pre-coding for the "dirty-paper" channel (a channel with interference known at the transmitter), using dither and Wiener estimation. Simon Litsyn helped in showing the existence of lattices which are "good for almost everything," which turned out to be a crucial element in the asymptotic optimality of nested lattice based coding schemes for general network

problems. Dave Forney provided insightful comments about Uri's work, and – after noticing that the zero-interference case resolves an open question about lattice decoding of Voronoi codes – summarized his interpretations under the multiple-meaning title "Shannon meets Wiener" (2002).

Emin Martinian and Greg Wornell contributed the idea of lattice codes with variable partition (for source coding with distortion side information at the encoder) during my Sabbatical at MIT in 2002–2003.

The work in my research group during the years 2003–2010 revealed two new exciting aspects of lattice codes. Yuval Kochman developed the modulo-lattice modulation technique for joint source-channel coding, and in particular, for bandwidth conversion (an idea proposed earlier in Zvi Reznic's PhD work). Tal Philosof discovered (during his PhD research with Uri Erez and myself) that lattice codes are stronger than random codes for the "doubly dirty" multiple-access channel.

Although the material had been there for quite a few years, it took some courage and encouragement to initiate this book project. The idea was thrown into the air during my visit at Andy Loeliger and Amos Lapidoth's groups at ETH, in the summer of 2008, and suggested again by Jan Østergaard during my visit at Aalborg University a couple of months later. Dave Forney gave me important comments and suggestions in the early stages of the writing, and I thank him for that. Tom Cover, whose book with Joy Thomas was a source of inspiration for many years, was kind enough to give me a few writing-style tips during my visit at Stanford in the summer of 2009.

Our research students in Tel Aviv University provided enormous help during the writing of this book. The chapter about lattice error exponents grew from extensive discussions with Amir Ingber. Sergey Tridenski and Arie Yeredor made specific contributions to the section on error exponents for Voronoi codebooks. Or Ordentlich and Uri Erez helped me shape the material about the existence of good lattices and nested lattices. My thanks are due to Yuval Domb, Eli Haim, Anatoly Khina, Adam Mashiach, Nir and Michal Palgy, Nir Weinberger and Yair Yona for many fruitful discussions; and to the students who participated in my "Lattices in information theory" course in the fall of 2011 for the valuable feedback.

Special thanks are due to my programming assistant Ilai Bistritz, whose good advice went much beyond the numerical work, graphs and illustrations that he contributed to this book.

During the work I received help and good advice from Ofer Amrani, Benny Appelbaum, Joseph Boutrus, Shosh Brosh-Weitz, Robert Calderbank (who gave me his class notes on coded modulation), Avner Dor, Moran and Tal Gariby, Michael Gastpar, Bo'az Klartag, Frank Kschischang, Stella Achtenberg, Tamas Linder, Bobak Nazer, Jan Østergaard, Dan Rephaeli, Kenneth Rose, Yaron Shany, Anelia Somekh-Baruch and Alex Vardy. (And I have surely missed here some important people who helped along the way.) Comments on early drafts of the book were kindly provided by Ling Cong, Adam Mashiach, Jan Østergaard, Danilo Silva, Shlomo Shamai and Yaron Shani, who also provided references and pointers.

This writing project could last forever without the constant attention and professional advice of my editors at Cambridge University Press, Phil Meyler, Sarah Marsh and Mia Balashova.

I was extremely happy when Bobak Nazer agreed to join me in writing the chapter about Gaussian networks; his deep understanding of the subject and clear writing style took this part of the book to a much higher level. Also the chapter about modulo-lattice modulation greatly benefitted from the collaboration in writing with Yuval Kochman.

Last but not least, I could not have survived these four long years of writing without the infinite love and patience of my wife Ariella and three children Kessem, Shoni and Itamar.

Notation

Lattices

Λ	lattice
G	generating matrix (columns are basis vectors)
$\det(\Lambda)$	lattice determinant
$\mathcal{P}_0$	fundamental cell
$\mathcal{V}_0$, $\mathcal{V}_\lambda$	fundamental Voronoi cell, Voronoi cell of lattice point λ
$V(\Lambda)$	cell volume
$\gamma(\Lambda)$	point density
$\bmod \Lambda$, $\bmod_{\mathcal{P}_0} \Lambda$, $\mathbf{x}/\Lambda$	modulo-lattice operations
$Q_\Lambda(\cdot)$	lattice quantizer
$Q_\Lambda^{(NN)}(\cdot)$	nearest-neighbor lattice quantizer
$d_{\min}$	minimum distance
$N_\Lambda(d)$	number of lattice points at distance d from the origin
$N_\Lambda(d_{\min})$	kissing number of Λ
$r_{\text{pack}}(\Lambda)$, $r_{\text{cov}}(\Lambda)$	packing radius, covering radius
$r_{\text{eff}}(\Lambda)$	effective radius
$\rho_{\text{pack}}(\Lambda)$, $\rho_{\text{cov}}(\Lambda)$	packing and covering efficiencies
$\sigma^2(\Lambda)$	second moment
$G(\Lambda)$	normalized second moment (NSM)
$\Gamma_q(\Lambda)$, $\Gamma_s(\Lambda)$	vector-quantizer granular gain, shaping gain
$P_e(\Lambda, \sigma^2)$	error probability (in the presence of AWGN)
$\mu(\Lambda, \sigma^2)$	volume to noise ratio (VNR)
$\mu(\Lambda, P_e)$	normalized volume to noise ratio (NVNR)
$\mu_{\text{matched}}(\Lambda, \mathbf{Z}, P_e)$	noise-matched NVNR
$\mu_{\text{euclid}}(\Lambda, \mathbf{Z}, P_e)$	Euclidean (mismatched) NVNR
$\mu_{\text{mix}}(\Lambda_1, \Lambda_2, P_e, \alpha)$, $\mu_{\text{mix}}(\Lambda_1, \Lambda_2, P_e, \xi)$	mixture-noise NVNR
$\Gamma_c(\Lambda, P_e)$	coding gain (relative to cubic lattice)
$\mathbf{U}$, $\mathbf{U}_{\text{eq}}$	dither, equivalent dithered quantization noise
R_{ECDQ}	entropy rate of lattice quantizer
$R_\infty(\Lambda)$	rate per unit volume

$\mathbb{L}$	Minkowski–Hlawka–Siegel (MHS) ensemble
$N_{\mathcal{S}}(\Lambda)$	number of non-zero lattice points in $\mathcal{S}$
J	nesting matrix
$\Gamma = \Gamma(\Lambda_1, \Lambda_2)$	nesting ratio
Λ_1/Λ_2	quotient group, relative cosets
$\mathcal{C}_{\Lambda_1,\mathcal{P}_0(\Lambda_2)}$, $\mathcal{C}_{\Lambda_1,\mathcal{V}_0(\Lambda_2)}$	lattice-shaped codebook, Voronoi codebook
$\mathcal{C}_{\mathbf{u},\Lambda_1,\mathcal{P}_0}$	dithered codebook
$R(\Lambda_1/\Lambda_2)$	codebook rate [bit per dimension]
$\mathbf{Z}_{\text{eq}}$	equivalent noise in mod Λ channel

Information theory

$H(X)$	regular entropy (of random variable X)
$H_B(p)$	binary entropy
$h(X)$	differential entropy
$I(X;Y)$	mutual information (between a pair of random variables)
$P_E(X)$	entropy power of a random variable ($P_E(X) = 2^{2h(X)}/2\pi e$)
C	channel capacity
C_∞	capacity per unit volume (Poltyrev's capacity)
$C^{(d)}$, $C^{(\text{euclid-th})}$	mismatched capacities
$R(D)$	rate-distortion function
$\mathcal{A}_\epsilon^{(n)}$	typical set
$\mathcal{C}$	codebook
$r_{\text{noise}} = \sqrt{n}\sigma^2$	typical AWGN radius

General

x, y	scalar variables
$\mathbf{x}, \mathbf{y}$	vector variables
X, Y	random variables
$\mathbf{X}, \mathbf{Y}$	random vectors (column form)
$\mathbf{X}^t$	$\mathbf{X}$ transpose (row form)
Var($\mathbf{X}$)	average variance per dimension
$\mathbb{R}^n$	Euclidean space
$\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$	integers
$\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$	modulo-q group
$N(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$\mathcal{B}_r$	a ball of radius r centered about the origin
$\mathcal{B}(\mathbf{x}, r)$	a ball of radius r centered at $\mathbf{x}$
V_n	volume of a unit-radius n-dimensional ball

$\mathrm{Vol}(S)$	volume of a set S
$\dot{=}$	equality to the first order in the exponent
$\otimes$	binary convolution ($p \otimes q = p(1-q) + q(1-p)$)
$[x]^+$	maximum between x and zero.

Abbreviations

AWGN	additive white-Gaussian noise
BPSK	binary phase-shift keying
BSC	binary-symmetric channel
BSS	binary-symmetric source
ECDQ	entropy-coded dithered quantization
MAC	multiple-access channel
ML	maximum likelihood
MSE	mean-squared error
NN	nearest-neighbor
NSM	normalized second moment
NVNR	normalized volume to noise ratio
PAM/QAM	pulse/quadrature-amplitude modulation
PAPR	peak to average power ratio
SER	symbol error rate
SNR	signal to noise ratio
VNR	volume to noise ratio
VNER	volume to noise entropy power ratio
VRDM	variable-rate dithered modulation

1 Introduction

Roughly speaking, a lattice is a periodic arrangement of points in the n-dimensional Euclidean space.[1] It reflects the "geometry of numbers" – in the words of the late nineteenth century mathematician Hermann Minkowski. Except for the one-dimensional case (where all lattices are equivalent up to scaling), there are infinitely many shapes of lattices in each dimension. Some of them are better than others.

Good lattices form effective structures for various geometric and coding problems. Crystallographers look for symmetries in three-dimensional lattices, and relate them to the physical properties of common crystals. A mathematician's classical problem is to pack high-dimensional spheres – or cover space with such spheres – where their centers form a lattice. The communication engineer and the information theorist are interested in using lattices for quantization and modulation, i.e., as a means for lossy compression (source coding) and noise immunity (channel coding). Although these problems seem different, they are in fact closely related.

The effectiveness of good lattices – as well as the complexity of describing or using them for coding – increases with the spatial dimension. Such lattices tend to be "perfect" in all aspects as the dimension goes to infinity. But what does "goodness" mean in dimensions 2, 3, 4, . . .?

In two dimensions, the *hexagonal lattice* is famous for the honeycomb shape of its Voronoi cells. The centers of the billiard (pool) balls in Figure 1.1 fall on a hexagonal lattice, which forms the tightest packing in two dimensions. The same hexagonal lattice defines a configuration for deploying cellular base stations that maximizes the coverage area per base station.

Interestingly, however, for higher dimensions the problems of packing and covering are *not* equivalent. In Figure 1.2, the centers of the oranges fall on the face-centered cubic (FCC) lattice, which is the best known sphere packing in three dimensions. In contrast, the best deployment of cellular base stations in a skyscraper (which maximizes their *three-dimensional* coverage) is over a body-centered cubic (BCC) lattice, illustrated in Figure 1.3.

[1] See the Wikipedia disambiguation page for other meanings of the word "lattice": in art and design, music, math and science.

Figure 1.1 Billiard (pool) balls packed in a triangle, for an initial game position.

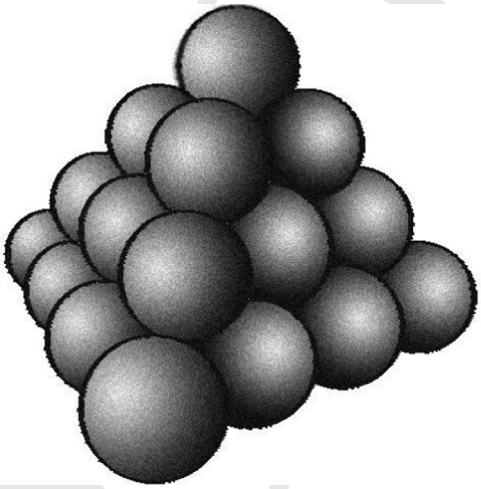


Figure 1.2 Packing oranges in a pile: each row is half-diameter shifted with respect to the previous row to reduce the unused volume. Similarly, each layer is staggered to fill the holes in the layer below it. The centers of the oranges form a lattice known as a face-centered cubic (FCC) lattice.

Which is the "best" lattice in each dimension is a question we shall not address; issues of efficient design and coding complexity of lattices are not at the focus of this book either. Instead, we characterize the performance of a lattice code by its thickness (relative excess coverage) and density (relative packed volume), and by the more communication-oriented figures of merit of normalized second moment (NSM) for quantization, and normalized volume to noise ratio (NVNR) for modulation. We define these quantities in detail in Chapter 3, and use them in Chapters 4–9 to evaluate lattice codes for the basic *point-to-point* source and channel coding problems. As we shall see, high-dimensional lattice codes can close

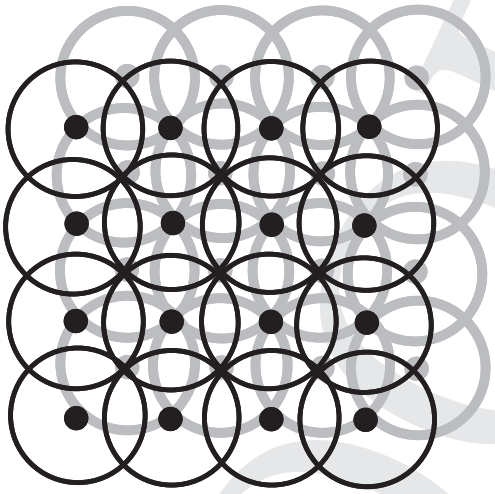


Figure 1.3 Three-dimensional sphere covering with a BCC lattice, describing the best deployment of cellular base stations in a skyscraper. The solid line shows even layers; the gray line shows odd layers. Compare the staggering pattern with that of the pile of oranges in Figure 1.2.

the gap to the information theoretic limits of communication: the capacity and rate-distortion function, quantities introduced by Shannon in his seminal 1948 paper [240], and further refined during the 1950s and 1960s.

The 1970s and 1980s saw the blooming of network information theory. Remarkably, some of the fundamental network problems were successfully solved using Shannon's information measures and *random coding* techniques, now with the additional variant of random binning. Simple examples of such network setups are *side-information* problems: the Slepian–Wolf and Wyner–Ziv source coding problem, and the Gelfand–Pinsker "dirty-paper" channel coding problem. The lattice framework provides a *structured coding* solution for these problems, based on a nested pair of lattices. This nested lattice configuration calls for new composite figures of merit: one component lattice should be a good channel code (have a low NVNR), while the other component lattice should be a good quantizer (have a low NSM). For joint source-channel coding problems, lattices with a good NSM-NVNR product are desired. We shall develop these notions in Chapters 10 and 11.

The curious reader may still wonder why we need a book about lattices in information theory. After all, Shannon's probabilistic measures and random coding techniques characterize well the limits of capacity (channel coding) and compression (source coding), and they also allow the study of source and channel networks [53, 64]. From the practical world side, communication theory provides ways to combine modulation with "algebraic" codes and approach the Shannon limits.

All this is true, yet between the theoretical and the constructive points of view something gets lost. Both the probabilistic and the algebraic approaches somewhat

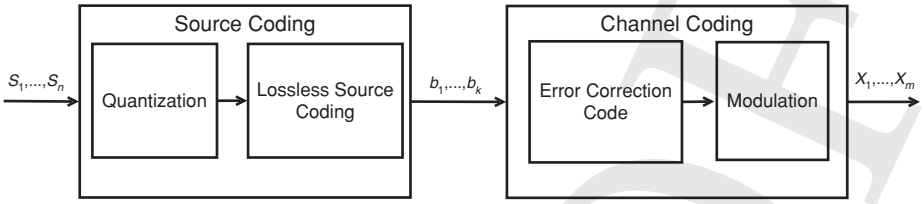


Figure 1.4 Source coding followed by channel coding. For an analog source and channel, the combined system maps a point in $\mathbb{R}^n$ (a source vector) to a point in $\mathbb{R}^m$ (a channel input vector). The ratio m/n is known as the "bandwidth-expansion factor."

hide the interplay between analog signals like sound or noise (created by nature) and digital modulation signals (created by man). Lattices are discrete entities in the analog world, and as such they bridge nicely the gap between the two worlds. At large dimensions, good lattices mimic the behavior of Shannon's random codes. For small dimensions, they represent an elegant combination of modulation and digital coding. As a whole, lattices provide a unified framework to study communication and information theory in an insightful and inspiring way.

Recent developments in the area of network information theory (mostly from the 2000s) have added a new chapter to the story of lattice codes. In some setups, structured codes are potentially *performance-wise better* than the traditional random coding schemes! And as Chapter 12 shows, the natural candidates to achieve the benefit of structure in Gaussian networks are, again, lattice codes.

1.1 Source and channel coding

Let us describe briefly how lattices fit into the framework of digital communication and classical information theory.

By Shannon's *separation principle*, transmission of an information source over a noisy channel is split into two stages: *source coding*, where the source is mapped into bits, and *channel coding*, where the digital representation of the source is mapped into a channel input signal. These two stages, which we describe in detail below, are illustrated in Figure 1.4.

The *source coding* (or compression) problem deals with compact digital representation of source signals. In *lossless* compression, our goal is to *remove redundancy* due to asymmetry in the frequency of appearance of source values, or to "memory" in the source. In this case, the source signal is available already in a digital form, say, as a sequence of binary symbols. And the task is to map n "redundant" source bits $\mathbf{s} = s_1, \ldots, s_n$ into $k = k(\mathbf{s})$ code bits, where $k < n$.[2]

[2] We would like k to be smaller than n for most source vectors (or for the most likely ones) in order to compress; but not too small, so the mapping would be invertible for (almost) all source vectors, for lossless reproduction.

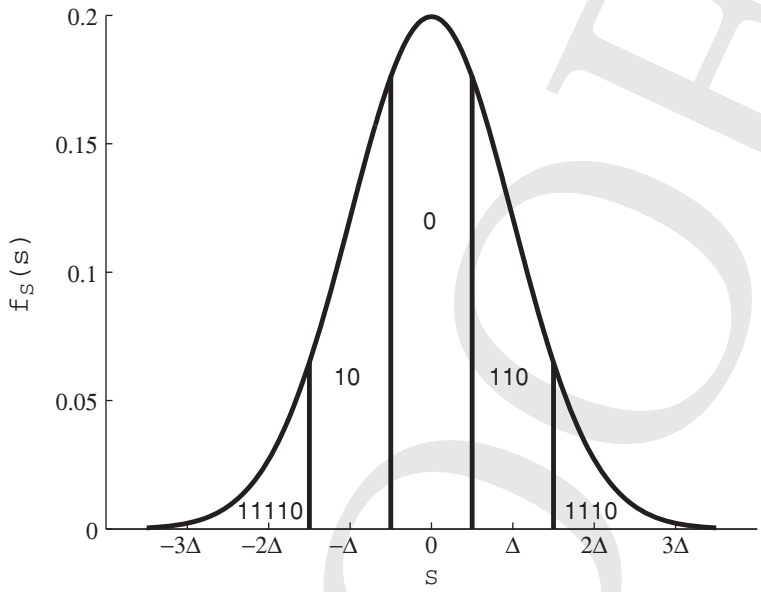


Figure 1.5 Scalar uniform quantization of a Gaussian source, followed by variable-length coding, i.e., $n = 1$ and k is varying. Each quantization level represents a range of source values.

In *lossy* compression, the source is usually *continuous* in nature: an analog representation of speech, sound, picture or video signal. Digitizing an analog signal consists first of converting it into a *discrete* form (both in time and in amplitude), and then coding it in the discrete alphabet domain. In discrete time the source is again a vector $\mathbf{s} = s_1, \ldots, s_n$, representing n consecutive source samples. After the vector $\mathbf{s}$ is encoded into a k-bit codeword, it is decoded and reconstructed as $\hat{\mathbf{s}} = \hat{s}_1, \ldots, \hat{s}_n$. The overall operation of mapping $\mathbf{s}$ to $\hat{\mathbf{s}}$ is called *quantization*, and the image (for a fixed k, the set of all 2^k possible reconstruction vectors $\hat{\mathbf{s}}$ in $\mathbb{R}^n$) is the quantization *codebook*.

A *lattice quantizer* codebook consists of points from an n-dimensional lattice. The codebook can be a truncated version (of size 2^k) of the lattice, or the whole lattice (with a variable codeword length $k = k(\hat{\mathbf{s}})$). We would like to make the bit rate $R = k/n$ (or the average coding rate $R = \bar{k}/n$) as *small* as possible, subject to a constraint on the reconstruction fidelity. Figure 1.5 shows the case of a scalar ($n = 1$) lattice quantizer with a variable code length $k(\hat{\mathbf{s}})$.

Channel coding deals with transmitting or storing information over a noisy channel or on a storage device. Our goal here is to *add redundancy* to the transmitted signal, to make it distinguishable from the noise. The channel input alphabet may be *discrete*, say, binary. In this case, transmission amounts to mapping k bits of information into n "redundant" code bits, where $n > k$.

The most common communication links are, however, over *continuous* media: telephone lines, cables or radio waves. The *baseband* channel representation is in

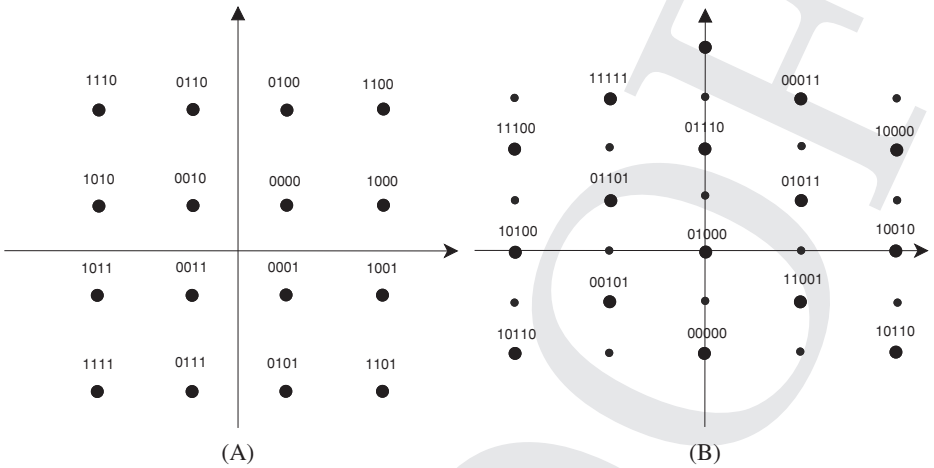


Figure 1.6 Two-dimensional finite lattice constellations, consisting of 16 points ($k = 4$). (A) A simple square constellation, representing uncoded quadrature-amplitude modulation (QAM); here $n' = k = 4$. (B) A hexagonal lattice constellation, represented as a mapping of redundant binary vectors of length $n' = 5$ into a rectangular constellation.

discrete time, so the channel input is a vector $\mathbf{x} = x_1, \ldots, x_n$. Coding over such a channel turns out to be in many ways the *dual* of encoding an analog source. It consists of two stages: an *error-correction coding* stage, where redundancy is added in the discrete alphabet domain (e.g., by converting k information bits to $n' > k$ code bits); and a *modulation* stage, where the digital codeword is mapped into the vector $\mathbf{x}$. The overall encoder mapping is thus of a k-bit information vector into a point in $\mathbb{R}^n$ (representing n consecutive channel inputs). The set of all 2^k possible input vectors $\mathbf{x}$ is called a codebook or a *constellation*.

A *lattice constellation* is a truncated version (of size 2^k) of an n-dimensional lattice. We would like to make the coding rate $R = k/n$ – which is now the (usually fixed) number of transmitted information bits per channel input – as *large* as possible, subject to a constraint on the probability of decoding error. See two examples of two-dimensional lattice constellations in Figure 1.6.

One benefit of the lattice coding framework that we can immediately recognize is that coding and modulation (or quantization) are combined as a *single entity*; a lattice code directly maps digital information (say, an index) into a vector in $\mathbb{R}^n$, and vice versa.

1.2 The information theoretic view

Information theory characterizes the ultimate performance limits of source and channel coding, as the code block length n goes to infinity.

In the channel coding case, the coding rate R is upper bounded (for a vanishing error probability) by the *Shannon capacity* C of the channel. The quantity C (associated with a memoryless channel with a transition distribution $p(y|x)$) is calculated by maximizing the mutual information (a functional of $p(x)$ and $p(y|x)$) over the input distribution $p(x)$. The maximizing input distribution $p^*(x)$ is used to prove the achievability of C: a set of $\approx 2^{nC}$ codewords is generated *randomly* and independently with an i.i.d. distribution $p^*(x)$; a *random coding* argument is then used to show that based on the channel output, the decoder can guess the correct transmitted codeword with a high probability as $n \to \infty$.

We see that *à la* Shannon, good codewords look like *realizations of random noise*. In the case of a binary-symmetric channel, the code generating noise consists of equally likely 0/1 bits. In the quadratic-Gaussian case, the code should be generated by a *white-Gaussian noise* (WGN).

Rate-distortion theory uses similar ideas to establish the ultimate performance limits of lossy source coding [18]. The Shannon *rate-distortion function* $R(D)$ lower bounds the coding rate R of any lossy compression scheme with distortion level of at most D (under some given distortion measure). And similarly to the channel coding case, computation of $R(D)$ induces an optimal reconstruction distribution, which is used to generate a good random codebook: independent realizations of a Bernoulli(1/2) sequence compose the codewords for a binary-symmetric source under Hamming distortion, while independent realizations of WGN compose the codewords for a white-Gaussian source under mean-squared distortion.

The fact that good codewords look like white noise is intriguing. Intuitively, one would expect the symbols of a codeword to be *dependent*, to distinguish them from the channel noise. This has made the random coding idea, on the one hand, a source of inspiration for many since Shannon presented his landmark theory in 1948. On the other hand, it sets a challenge for finding more *structured* ways to approach the information theoretic limits, ways in which the dependence between the code symbols is more explicit. Can noise be realized in a structured way?

1.3 Structured codes

The Hamming code – mentioned already in Shannon's 1948 paper – was the early bird of the structured coding approach. It was followed by the breakthrough of algebraic coding theory in the 1950s and 1960s [21]. The implication was that, in fact, a good collection of random-like bits can be constructed as an additive group in the binary modulo-2 space. These *linear codes* take various forms, such as Reed–Muller, BCH and, more recently, LDPC, turbo and polar codes, and they also have extensions to non-binary (Reed–Solomon) codes and convolutional (trellis) codes.

Common to all these codes is that for a random message, the resulting n-length codeword is indeed roughly uniformly distributed over the n-dimensional binary

space. That is, each code bit takes the values 0 and 1 with equal probability; furthermore, small subsets of code bits are roughly independent.

The extension of this concept to continuous signals is however less obvious: can a code mimic Gaussian noise in a structured way? A first step towards this goal is provided by Shannon's *asymptotic equipartition property* (AEP). In a high dimension n, the *typical set* of WGN of variance σ^2 is a spherical shell of radius $\approx\sqrt{n\sigma^2}$. Thus, the codewords of a good code are roughly uniformly distributed over such a spherical shell.

The concept of *geometrically uniform codes* (GUC) [86] suggests a deterministic characterization for a "uniform-looking" code: every codeword should have the same *distance spectrum* to its neighboring codewords. This concept captures the desired property of a good Euclidean code, in both the block and the convolutional (*trellis*) coding frameworks.

Due to their periodic and linear structure, lattices are natural candidates for *unbounded* GUCs. For example, the commonly used QAM constellation shown in Figure 1.6(A) is a truncated version of the *square lattice*, while the more "random-like" set of two-dimensional codewords shown in Figure 1.6(B) is a truncated version of the *hexagonal lattice*. Moreover, the code designer can shape the borders of these constellations to be more round, for example, by truncating them into a circle or into a coarser hexagonal cell. And as the dimension gets high, lattices which are truncated into a "good" coarse lattice cell become closer to a randomly generated Gaussian codebook.

1.4 Preview

We shall get to the exciting applications mentioned earlier after building up some necessary background. The book starts by introducing lattices in Chapter 2, and the notions of lattice goodness in Chapter 3. Chapter 4 introduces two central players in our framework: dithering, which is a means to randomize a lattice code, and Wiener estimation, which is a means to reduce the quantization or channel noise. The importance of these techniques will be revealed gradually throughout the book.

Equipped with these notions and techniques, we consider variable-rate ("entropy-coded") dithered quantization (ECDQ) using an *unbounded* lattice in Chapter 5. In particular, we shall see how the NSM characterizes the redundancy of the ECDQ above Shannon's rate-distortion function. The reader who is interested primarily in channel coding may skip Chapter 5, and continue directly to modulation with an unbounded lattice constellation in Chapter 6.[3] This chapter shows how the NVNR determines the gap from capacity of a lattice constellation. It also describes variable-rate dithered modulation, which is the channel coding counterpart of ECDQ.

[3] Sections which are optional reading for the flow of the book are denoted by an asterisk.

Before moving to more advanced coding setups, we stop to examine the existence of asymptotically good lattices in Chapter 7. In Chapter 8 we define nested lattices, and *finite* Voronoi-shaped codebooks taken from a lattice. These notions form in Chapter 9 the basis for Voronoi modulation, which achieves the capacity of a power-constrained AWGN channel, and for Voronoi quantization, which achieves the quadratic-Gaussian rate-distortion function. In both these solutions, dither and Wiener estimation play crucial roles.

A small step takes us from the point-to-point communication setups above to side-information problems in Chapter 10. We shall construct lattice code solutions for the Wyner–Ziv problem (source coding with side information at the decoder) and the "dirty-paper" problem (channel coding with side information at the encoder). These lattice coding schemes serve as building blocks for common multi-terminal communication problems: encoding of distributed sources and broadcast channels. Before moving to more general networks, we examine in Chapter 11 a lattice-based joint source-channel coding technique, called modulo-lattice modulation (MLM). A combination of MLM and prediction leads to "analog matching" of sources and channels with mismatched spectra, and to "bandwidth conversion." Chapter 12 extends the discussion on multi-terminal problems to general Gaussian networks. There we shall see that when side information is distributed among several nodes of the network, lattice codes are not only attractive complexity-wise, but sometimes they have better performance than traditional random coding and binning techniques.

Chapter 13 complements the discussion of asymptotically good lattice codes in Chapter 7 by examining their error exponents. As for capacity, good lattice codes turn out to be optimal also in terms of this more refined aspect.

Information theory is not a critical prerequisite for reading this book, but (starting from Chapter 5) we use information measures, such as entropy, mutual information and capacity, to assess system performance. To keep the book self-contained, the Appendix includes elementary background in information theory, as well as some other complementary material.

As mentioned above, dithering and Wiener estimation are central concepts in the lattice coding framework. The question of where and in what sense they are necessary will follow our discussion throughout the book.

What's *not* in the book?

The writer has the freedom to focus on his favorite subject. Naturally (in the case of this writer) the book takes an information theoretic flavor, with less emphasis on coding theoretic aspects. For algebra of lattices, and for specific constructions of lattices and coded-modulation schemes from error-correcting codes, the reader is referred to the comprehensive book of Conway and Sloane [49], and to the excellent class notes of Forney [81] and Calderbank [28].

Encoding and decoding complexity is a topic of theoretical as well as practical importance, although traditionally neglected by information theory. A good introduction to the subject can be found in the survey paper of Agrell *et al.* [3]. The vast literature on MIMO communication contains numerous publications about the design of linear coded-modulation schemes and efficient lattice decoding algorithms.

In the fight between a timely manuscript and time of publication, some topics which are natural to the spirit of the book were left out. One such topic is the extension to *colored*-Gaussian sources and channels; see, for example, [211, 288, 291]. Another topic is the emerging area of lattice wiretap codes; see, for example, the survey paper by Liang *et al.* [156] and other recent work [118, 168]. Hopefully these topics will find their way to a later edition of the book.

Finally, since the late 1990s lattice-based cryptography has been a major area of research in computer science. Its connection to lattice codes for communication is yet to be explored; see the book by Micciancio and Goldwasser [186], and the survey by Micciancio and Regev [188].

2 Lattices

The simplest lattice is the one-dimensional grid $\{\ldots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \ldots\}$. In one dimension, all lattices are equivalent up to scaling. To make life more interesting – and to obtain better geometric properties – we must consider multi-dimensional lattices.

This chapter presents n-dimensional lattices and important ideas associated with lattice codes that are used throughout the book. We take a geometric and, for some asymptotic results, probabilistic viewpoint. The algebraic aspects of lattices – although crucial for their implementation at a low complexity – are secondary for our purposes, and will not be treated in this book.

We restrict our attention to communication problems in which the lattice code is selected by the system designer. Thus, we rely on the existence of lattices with certain "good" properties, and on algorithms for encoding and decoding them at a reasonable complexity. [1]

We start with the basic definitions of a lattice and lattice partition.

2.1 Representation

A lattice is a regular array in the Euclidean space. Mathematically, it is a *discrete sub-group* of $\mathbb{R}^n$: a set of points which is *closed under reflection and real addition*. The set is *discrete* in the sense that the distance between any two points is greater than some positive number. If a point λ is in the lattice then so is its reflection $-\lambda$, and if two points λ_1 and λ_2 are in the lattice then so is their vector sum $\lambda_1 + \lambda_2$. Thus, the origin (the point $\mathbf{0}$) is always in the lattice because it is the sum of λ and $-\lambda$.

[1] The situation is different when the lattice is selected by nature or at random. For example, in digital communication (e.g., QAM) over a fading MIMO channel, the physical multi-path channel behaves like a random matrix which creates an equivalent lattice constellation at the receiver. In cryptography, a "hard-to-break" lattice is created by a random number generator.

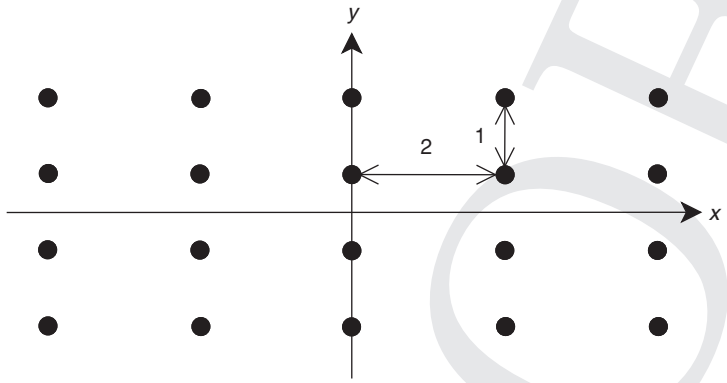


Figure 2.1 The two-dimensional grid $\{(2i, j) : i, j \in \{0, \pm 1, \pm 2, \ldots\}\}$ contains all points in the plane whose y-coordinate is an integer and whose x-coordinate is an integer multiple of 2.

Furthermore, the lattice is a countably infinite set: it must contain all integer multiples $\pm 2\lambda, \pm 3\lambda, \pm 4\lambda, \ldots$ of any lattice point λ, as well as all integer linear combinations $\lambda_1 \pm \lambda_2, \lambda_1 \pm 2\lambda_2, \ldots, 3\lambda_1 \pm 2\lambda_2, \ldots$, of any two lattice points λ_1 and λ_2, etc.

We can obtain simple multi-dimensional lattices by taking the Cartesian product of scalar lattices, like the two-dimensional grid shown in Figure 2.1. Such a simple grid, however, would not allow us to obtain the efficient arrangements of oranges and cellular base stations shown in Figures 1.2 and 1.3. Our next step is to define a lattice in a more general and constructive way.

The linearity property of the lattice reminds us of a linear vector space. It is only in the latter that any real-valued coefficients, and not just integer multiples, are possible. This analogy calls for a definition of a lattice in terms of a *basis*.

Definition 2.1.1 *A non-degenerate n-dimensional lattice Λ is defined by a set of n linearly independent basis (column) vectors $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$ in $\mathbb{R}^n$. The lattice Λ is composed of all integral combinations of the basis vectors, i.e.,*

$$\begin{aligned} \Lambda &= \left\{ \lambda = \sum_{k=1}^{n} i_k \boldsymbol{g}_k \ : \ i_k \in \mathbb{Z} \right\} \\ &= \left\{ \lambda = G \cdot \mathbf{i} \ : \ \mathbf{i} \in \mathbb{Z}^n \right\}, \end{aligned} \tag{2.1}$$

where $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ is the set of integers, $\mathbf{i} = (i_1, \ldots, i_n)^t$ is an n-dimensional integer (column) vector, and the $n \times n$ generator matrix *G is given by*

$$G = [\, \boldsymbol{g}_1 \,|\, \boldsymbol{g}_2 | \ \ldots \ | \, \boldsymbol{g}_n \,].$$

The resulting lattice is denoted $\Lambda(G)$.

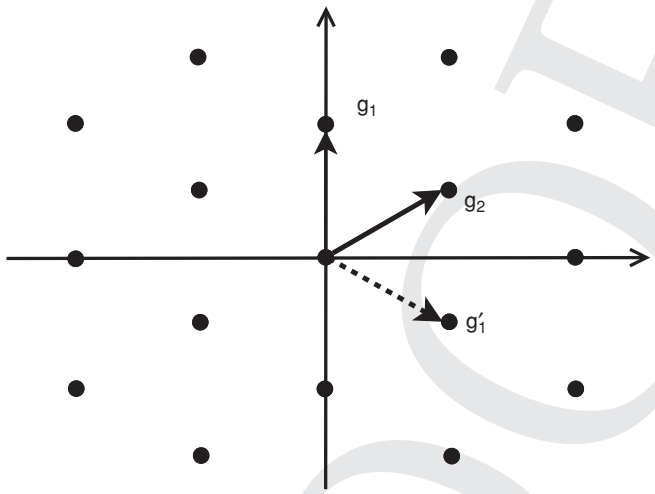


Figure 2.2 The hexagonal lattice is generated by the basis vectors $\mathbf{g}_1 = (0, 2)^T$ and $\mathbf{g}_2 = (\sqrt{3}, 1)^T$. But it can also be generated by the pair $\mathbf{g}'_1 = (\sqrt{3}, -1)^T$ and the same $\mathbf{g}_2$, or by the pair $\mathbf{g}''_1 = (\sqrt{3}, 1)^T$ and $\mathbf{g}''_2 = (-\sqrt{3}, 1)^T$. Clearly this lattice cannot be written as a Cartesian product of two scalar lattices. Nevertheless, we can construct it by alternating between two staggered horizontal scalar lattices, one for the even rows and one, half-step shifted, for the odd rows.

Figure 2.2 shows the famous two-dimensional hexagonal lattice – denoted as A_2. The reason why it is called "hexagonal" will become clear in the next section.

We shall soon discuss the degenerate case, where the number of basis vectors in G is less than the dimension n, or the basis vectors are linearly dependent. When G is an *identity matrix*, we get the *integer lattice*, $\Lambda = \mathbb{Z}^n$, also called the *cubic lattice* or "$\mathbb{Z}$ lattice." Any lattice can be viewed as a linear transformation, by the generator matrix, of the integer lattice:

$$\Lambda = G \cdot \mathbb{Z}^n, \tag{2.2}$$

which is simply another way of writing (2.1).

However, the generator matrix is not unique for a given lattice. A lattice is invariant to a *unimodular transformation* of its basis.

Proposition 2.1.1 (Change of basis) *A matrix G' generates the same lattice as G, i.e., $\Lambda(G') = \Lambda(G)$, if and only if*

$$G' = G \cdot T = [\, G\mathbf{t}_1 \mid G\mathbf{t}_2 \mid \ldots \mid G\mathbf{t}_n \,] \tag{2.3}$$

for some unimodular matrix $T = [\, \mathbf{t}_1 \mid \mathbf{t}_2 \mid \ldots \mid \mathbf{t}_n \,]$*, i.e., an integer matrix with a unit absolute determinant,* $\det(T) = \pm 1$.

Proof If T satisfies the condition, then each column of G' is an integer combination of the columns of G, i.e., $\boldsymbol{g}'_j = G\mathbf{t}_j = \sum_{i=1}^{n} t_{ij}\boldsymbol{g}_i$. Thus, by Definition 2.1.1, $\Lambda(G')$

is contained in $\Lambda(G)$. Conversely, since $\det(T) = \pm 1$, the inverse matrix T^{-1} is a (unit-determinant) integer matrix too (by Cramer's rule for matrix inversion), so $\Lambda(G')$ also contains $\Lambda(G)$. Hence, $\Lambda(G')$ and $\Lambda(G)$ must be identical. To prove the "only if" part, note that since the basis vectors are linearly independent, T must be integer valued otherwise $\Lambda(TG)$ will contain points outside $\Lambda(G)$. The same argument shows that if $|\det(T)|$ is greater than 1, then $|\det(T^{-1})|$ is smaller than 1, hence $\Lambda(T^{-1}G')$ contains points outside $\Lambda(G')$. Thus $\Lambda(G') = \Lambda(G)$ implies $|\det(T)| = 1$. □

A by-product of Proposition 2.1.1 is that all (square) generator matrices of a lattice have the same absolute determinant: $\det(G') = \det(GT) = \det(G)\det(T) = \pm\det(G)$. Thus, the absolute value of the determinant of the generator matrix is an invariant property of the lattice.

Definition 2.1.2 (Lattice determinant) [2] *The lattice determinant* $\det(\Lambda)$ *is defined as the absolute determinant of its generator matrix* $|\det(G)|$.

Due to the linear independence of the basis vectors, the matrix G is non-singular, thus $\det(\Lambda) > 0$.

As we saw in Figure 2.1, a simple way to construct high-dimensional lattices is by taking Cartesian products of lower-dimensional lattices.

Definition 2.1.3 (Cartesian product) *The Cartesian product of two lattices* Λ_1 *and* Λ_2 *of dimensions* n_1 *and* n_2 *is an* $n = n_1 + n_2$ *dimensional lattice:*

$$\Lambda_1 \times \Lambda_2 = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} : \mathbf{x} \in \Lambda_1, \mathbf{y} \in \Lambda_2 \right\}. \tag{2.4}$$

The generator matrix of the product lattice is a block-diagonal matrix

$$G = \begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix} \tag{2.5}$$

with the component generator matrices on its diagonal, hence its determinant is the product of the component determinants $\det(\Lambda_1 \times \Lambda_2) = \det(\Lambda_1) \cdot \det(\Lambda_2)$.

Equivalent dimension We expect that under a "natural" goodness measure, the product lattice $\Lambda \times \cdots \times \Lambda$ is as good as its component lattice Λ; hence, both lattices have the same *equivalent dimension*.

2.1.1 Characterization of lattice bases

Does the invertible generator matrix form (2.1) describe the most general arrangement of points satisfying the linearity property at the beginning of the section?

[2] In the literature (e.g., [49]) $\det(\Lambda)$ is sometimes defined as $\det^2(G)$, which is also the determinant of the *Gram matrix* $G^t G$.

Degenerate and dense lattices A lattice in $\mathbb{R}^n$ may have *less* than n basis vectors (or the basis vectors may be linearly dependent). In this case, the lattice is contained in a linear sub-space of $\mathbb{R}^n$, and is called *degenerate*. For example, the basis vectors $[1, -1, 0]^t$ and $[0, 1, -1]^t$ span a two-dimensional hexagonal lattice, which is embedded in some tilted plane in $\mathbb{R}^3$. See Problem 2.1.

On the other hand, we never need more than n basis vectors to generate a lattice in $\mathbb{R}^n$: if some of the vectors are linearly dependent then, either a smaller basis for the lattice can be found, or the set generated by (2.1) is *dense* (non-discrete) and therefore cannot be considered as a lattice.

Example 2.1.1 (Dense lattice) *In one dimension, $G = (1, \sqrt{2})$ generates a* dense *set; that is, integer combinations of* 1 *and* $\sqrt{2}$ *can arbitrarily approach any point in* $\mathbb{R}$. [3]

Example 2.1.2 (Extended basis) *A basis is not necessarily a subset of an extended basis. In one dimension, the points* 9 *and* 10 *span the entire* $\mathbb{Z}$ *lattice, but none of them can span it alone. In two dimensions, the three points* (1,2)*,* (2,1) *and* (2,2) *span the entire* $\mathbb{Z}^2$ *lattice, but neither pair does.*

Primitive points A lattice point λ is called *primitive* if it is the shortest lattice point in its direction, i.e., $\alpha\lambda$ is *not* in Λ, for all $0 < \alpha < 1$. Basis vectors are necessarily primitive, but the opposite is not true.

Example 2.1.3 (Checkerboard lattice) *A set of n linearly independent primitive vectors does* not *necessarily form a basis for a lattice. Consider as an example the n-dimensional "checkerboard" lattice, which consists of all the all-even and all-odd vectors in* $\mathbb{R}^n$*, i.e., the union of* $2\mathbb{Z}^n$ *and* $[1, \ldots, 1] + 2\mathbb{Z}^n$*. Figure* 2.3 *shows the two-dimensional case. In three dimensions, this is exactly the BCC lattice of Figure* 1.3*. A simple basis for this lattice consists of the all-one vector* $[1, \ldots, 1]$*, plus any* $n - 1$ *vectors from the set of n elementary even vectors* $[2, 0, \ldots, 0], \ldots, [0, \ldots, 0, 2]$*. Note that the elementary even vectors are primitive, independent of each other, and, for* $n > 4$*, shorter than the all-one vector. However, they cannot span odd vectors; hence, without the all-one vector they do not form a basis for the checkerboard lattice. (See for comparison the definition of the* D_n *lattice in Example* 2.4.2*.)*

Good basis for a given lattice Since the basis is not unique, we may ask which basis is "best" for a given lattice. The answer is, however, not precise. A common rule of thumb for a good basis is that

- the basis vectors $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$ are the shortest possible,
- the basis vectors are nearly orthogonal.

[3] *Quasicrystals* can be modeled using a basis with more than three vectors in $\mathbb{R}^3$ [158].

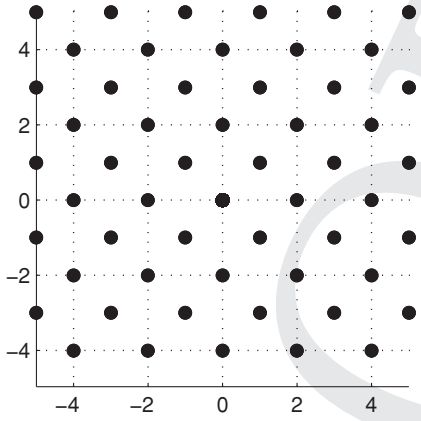


Figure 2.3 Checkerboard lattice in two dimensions.

The first criterion guarantees numerical stability, while the second is useful for reducing the *complexity* of searching for the *closest lattice point* to a given point in space – a problem which is at the heart of the coding and decoding of lattice codes. It nevertheless turns out that the two criteria are closely related by the Hadamard inequality [53]:

$$\det(\Lambda) = |\det(G)| \leq \prod_{i=1}^{n} \|\boldsymbol{g}_i\| \tag{2.6}$$

with equality if and only if the basis vectors are *orthogonal*. Thus, a short basis also tends to be close to orthogonal. The *LLL algorithm* [154] reduces a given basis into a new and usually shorter basis, which satisfies a certain "near-orthogonality" criterion.

Interestingly, the *n shortest* lattice vectors do *not* necessarily form a basis. Take, for example, the checkerboard lattice (Example 2.1.3): the elementary even vectors do not form a basis for that lattice, although for dimension $n > 4$ they are the shortest (in particular, shorter than the all-one vector, whose length is $\sqrt{n}$).

2.1.2 Cosets

The final point we should discuss before the end of this section is that of a lattice shift, or *coset*, defined as

$$\Lambda_{\mathbf{x}} = \mathbf{x} + \Lambda = \{\mathbf{x} + \lambda : \lambda \in \Lambda\}. \tag{2.7}$$

A coset is a discrete set of points such that the difference vector between every pair of points belongs to the lattice. However, the coset itself is, in general, *not* a lattice, as it is not closed under reflection and addition; in particular, it does not contain the origin.

Clearly, the union of $\Lambda_{\mathbf{x}}$ over all shifts $\mathbf{x}$ covers the entire space $\mathbb{R}^n$. But this union contains many overlaps. A natural question to ask then is: what is the *minimal set* of shifts S such that

$$\bigcup_{\mathbf{x}\in S} \Lambda_{\mathbf{x}} = \mathbb{R}^n \ ? \tag{2.8}$$

This question leads us to the subject of *lattice partition*.

2.2 Partition

A lattice induces a division of the Euclidean space into *congruent cells*. Like the lattice representation, this division is not unique; there are many ways to partition space with respect to a given lattice Λ.

From a geometric viewpoint, the most important division is the *Voronoi* partition, which uses a *nearest-neighbor* (NN) rule. Let $\|\cdot\|$ denote some norm, for example, Euclidean distance. The distance of a point $\mathbf{x}$ in $\mathbb{R}^n$ from Λ is defined as

$$\|\mathbf{x} - \Lambda\| \stackrel{\Delta}{=} \min_{\lambda\in\Lambda} \|\mathbf{x} - \lambda\|. \tag{2.9}$$

The *nearest-neighbor quantizer* $Q_\Lambda^{(NN)}(\cdot)$ maps $\mathbf{x}$ to its closest lattice point:

$$Q_\Lambda^{(NN)}(\mathbf{x}) = \arg\min_{\lambda\in\Lambda} \|\mathbf{x} - \lambda\|, \tag{2.10}$$

and the *Voronoi cell* $\mathcal{V}_\lambda$ is the set of all points which are quantized to λ:

$$\mathcal{V}_\lambda = \{\mathbf{x} : \ Q_\Lambda^{(NN)}(\mathbf{x}) = \lambda\}. \tag{2.11}$$

The breaking of ties in (2.10) is carried out in a systematic manner, so that the resulting Voronoi cells $\{\mathcal{V}_\lambda, \lambda \in \Lambda\}$ are congruent.

If not stated otherwise, the Voronoi partition refers to using the Euclidean norm in (2.9) and (2.10). In this case, the Voronoi cell $\mathcal{V}_\lambda$ is a convex polytope, which – like the lattice – is symmetric about the origin. See Problem 2.2. Each face of $\mathcal{V}_\lambda$ is determined by a hyperplane, crossing orthogonally to the line connecting λ to one of its neighbors. These neighbors are then called *face-determining points*.

Example 2.2.1 (Honeycomb) *The Voronoi partition of the lattice of Figure* 2.2 *(with* $G = \left(\begin{smallmatrix} 0 & \sqrt{3} \\ 2 & 1 \end{smallmatrix}\right)$*) divides the plane into equilateral hexagonal cells with edge length* $2/\sqrt{3}$*, as shown in Figure* 2.4(A)*. A possible "tie breaking" rule, which keeps the cells congruent, is that each cell contains three out of its six edges and two out of its six corners, with the* same *orientation for all cells.* [4]

[4] Any systematic association of *half* the (non-corner) boundary points to each cell would keep the cells congruent. This is because each of these points is on the border of two cells, while each corner point is on the border of three cells. For n-dimensional cells, boundary points are classified into n types of

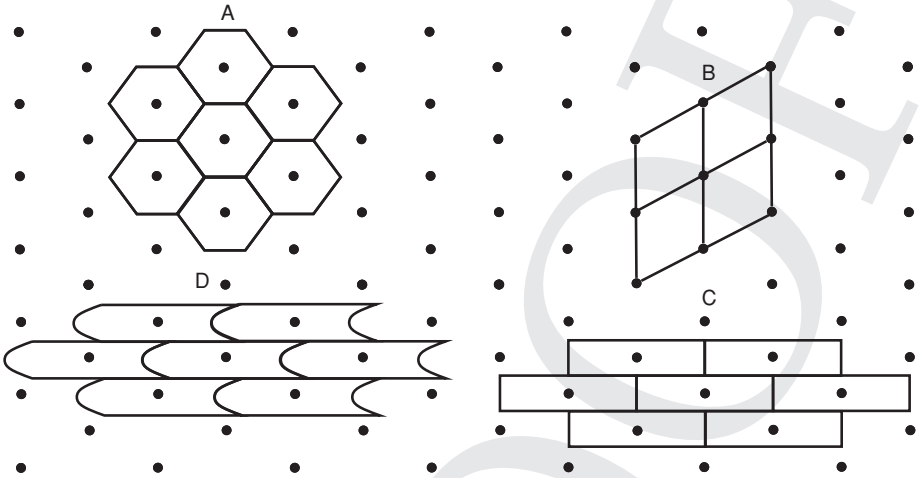


Figure 2.4 Hexagonal lattice with four possible partitions: (A) Voronoi partition; (B) parallelepiped partition; (C) "brick wall" partition generated by successive quantization (first quantize the y-component then, conditioned on that, quantize the x-component); (D) a general (non-polytope) fundamental cell.

The *fundamental* Voronoi cell $\mathcal{V}_0$ is the Voronoi cell associated with the origin ($\lambda = \mathbf{0}$). Due to the periodic nature of the lattice, all the Voronoi cells are shifted versions (by the lattice points) of $\mathcal{V}_0$. Hence, any point in space can be uniquely expressed as the sum of a lattice point and a point in the fundamental Voronoi cell.

As mentioned previously, we do not have to use the Euclidean distance in (2.10). A periodic partition will result by using *any* function of the difference $\mathbf{x} - \lambda$; an example comparing the ℓ_2 and ℓ_4 norms is shown in Figure 2.5. An alternative definition for a general lattice-based partition, which does not rely explicitly on a distance measure, is based on the notion of a *fundamental cell*.

We say that a collection of sets $\{S_i\}$ *covers* the Euclidean space if any point in space is in one of the sets, i.e., $\cup_i S_i = \mathbb{R}^n$. We say that the sets are *packed* in the Euclidean space if no point in space belongs to more than one set, i.e., $S_i \cap S_j = \emptyset$ for all $i \neq j$. Finally, if the sets both cover $\mathbb{R}^n$ *and* are packed in $\mathbb{R}^n$, then $\{S_i\}$ is a *partition* of $\mathbb{R}^n$.

Definition 2.2.1 (Fundamental cell, lattice partition) *A fundamental cell $\mathcal{P}_0$ of a lattice Λ is a bounded set, which, when shifted by the lattice points, generates a partition $\mathcal{P} = \{\mathcal{P}_\lambda\}$ of $\mathbb{R}^n$. That is,*

(i) each cell $\mathcal{P}_\lambda$ is a shift of $\mathcal{P}_0$ by a lattice point $\lambda \in \Lambda$

$$\mathcal{P}_\lambda = \mathcal{P}_0 + \lambda = \{\mathbf{x} : \ (\mathbf{x} - \lambda) \in \mathcal{P}_0\};$$

k-dimensional edges, for $k = 0, 1, \ldots, n-1$. Although the boundary has zero volume, its association to the cell is critical for lattice codebooks (see Chapter 9).

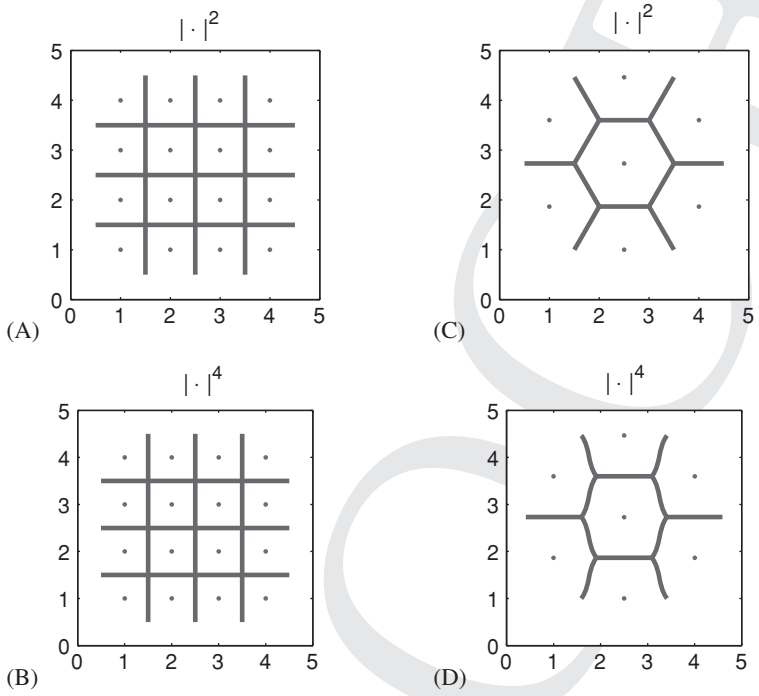


Figure 2.5 Examples of lattices and lattice partitions: (A) the $\mathbb{Z}^2$ lattice with a Euclidean Voronoi partition; (B) the $\mathbb{Z}^2$ lattice with a fourth-power norm-based Voronoi partition; (C) a hexagonal lattice with a Euclidean Voronoi partition; (D) a hexagonal lattice with a fourth-power norm-based Voronoi partition.

(ii) the cells do not intersect, $\mathcal{P}_\lambda \cap \mathcal{P}_{\lambda'} = \emptyset$ for all $\lambda' \neq \lambda$; and
(iii) the union of the cells covers the whole space, $\bigcup_{\lambda\in\Lambda} \mathcal{P}_\lambda = \mathbb{R}^n$.

It is convenient to think of a fundamental cell as a connected region, although the definition does not require that.

Definition 2.2.1 implies that given a lattice Λ and a fundamental cell $\mathcal{P}_0$, any point $\mathbf{x}$ in space can be *uniquely* expressed as a sum

$$\mathbf{x} = \lambda + \mathbf{x}_e, \quad \text{where } \lambda \in \Lambda \text{ and } \mathbf{x}_e \in \mathcal{P}_0. \tag{2.12}$$

We may think of λ in (2.12) as the *quantization* of $\mathbf{x}$ to the lattice Λ,

$$\lambda = Q_\Lambda(\mathbf{x}), \tag{2.13}$$

and of $\mathbf{x}_e$ in (2.12) as the *quantization error*. This extends the notion of a nearest-neighbor quantizer (2.10) with Voronoi partition (2.11), to the case of a general fundamental cell $\mathcal{P}_0$ inducing a lattice partition $\mathcal{P} = \Lambda + \mathcal{P}_0$.

The Voronoi partition generated by the nearest-neighbor quantizer (2.10) clearly satisfies the properties in Definition 2.2.1 (provided that ties are broken in a

systematic manner). The simplest lattice partition is, however, a *parallelepiped partition* generated by some lattice basis $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$. Here $\mathcal{P}_0$ is the *fundamental parallelepiped*, consisting of all points which are linear combinations of the basis vectors with coefficients between zero and one:

$$\mathcal{P}_0 = \left\{ \mathbf{x} = \sum_{i=1}^{n} \alpha_i \boldsymbol{g}_i : \ 0 \le \alpha_1, \ldots, \alpha_n < 1 \right\} \tag{2.14}$$

$$= G \cdot \text{Unit Cube}, \tag{2.15}$$

where Unit Cube $= \{\mathbf{x} : \ 0 \le x_i < 1, i = 1, \ldots, n\}$. Note that the unit cube is the parallelepiped partition of the $\mathbb{Z}$-lattice. [5] See Figure 2.4(B).

Since the lattice has more than one basis, its parallelepiped partition is not unique. Moreover, a shift or reflection of a fundamental cell is another fundamental cell, which generates another partition of the lattice. Interestingly, though, it follows from a simple "volume preservation" argument that the volume of a cell is the same under *any* lattice partition. And, as we shall see later, all lattice partitions are, in fact, *equivalent* in several senses; for example, any fundamental cell is a complete set of coset shifts in (2.8).

Proposition 2.2.1 (Cell volume) *The cell volume*

$$V = \text{Vol}(\mathcal{P}_0) = \int_{\mathcal{P}_0} d\mathbf{x} \tag{2.16}$$

is independent of the lattice partition $\mathcal{P}$, and it is equal to the lattice determinant of Definition 2.1.2

$$V = \det(\Lambda) = |\det(G)| \stackrel{\Delta}{=} V(\Lambda). \tag{2.17}$$

Proof Consider first the parallelepiped partition (2.14) induced by the generator matrix G. By a change of variables $\mathbf{x} = G\mathbf{x}'$, and using (2.15), we have

$$V = \int_{\mathcal{P}_0} d\mathbf{x} = |\det(G)| \int_{\text{Unit Cube}} d\mathbf{x}' = |\det(G)|. \tag{2.18}$$

Next, for a general partition, consider the cells contained in a large cube B. Since the cells have a finite diameter, the volume of the fractional cells at the boundary of the cube B becomes negligible when B is sufficiently large. Thus, if there are $N(B)$ lattice points inside B, then the cell volume is roughly

$$V \approx \frac{\text{Vol}(B)}{N(B)}, \tag{2.19}$$

independent of the shape of the cells, and this approximation becomes exact when the edge length of B, and hence also $N(B)$, go to infinity. □

[5] To see that the parallelepiped cell $\mathcal{P}_0$ in (2.14) satisfies the conditions of Definition 2.2.1, note that (i) the difference between any two points in $\mathcal{P}_0$ is *not* a lattice point, and (ii) every point outside $\mathcal{P}_0$ can be written as a sum of a point in $\mathcal{P}_0$ and a lattice point. See Lemma 2.3.2 in the next section.

In the following section (see Corollary 2.3.1) we shall see an alternative (more direct) proof for the second part of the proof above, showing that the cell volume is partition invariant.

The approximation in (2.19) holds, in fact, for any body which is large compared to the cells, i.e., the number of lattice points $N(S)$ in a large body S is approximately $\mathrm{Vol}(S)/V(\Lambda)$. We thus define the lattice *point density* as the reciprocal of the cell volume:

$$\gamma(\Lambda) = \frac{1}{V(\Lambda)}, \tag{2.20}$$

measured in *points per unit volume*.

2.3 Equivalent cells and coset leaders

An even stronger notion of equivalence between partitions holds: all the fundamental cells of a lattice are identical modulo a fixed partition. More explicitly, any fundamental cell can be decomposed into pieces and rearranged (via lattice shifts) to form another fundamental cell. Although this may seem to be a geometric property, it is, in fact, a consequence of the lattice being a sub-group of the Euclidean space.

Definition 2.3.1 (Mod $\mathcal{P}_0$, Mod Λ) *For a given lattice partition $\mathcal{P}$ with a fundamental cell $\mathcal{P}_0$, the* modulo fundamental cell *operation is defined as*

$$\mathbf{x} \bmod \mathcal{P}_0 = \mathbf{x}_e = \mathbf{x} - Q_\Lambda(\mathbf{x}), \tag{2.21}$$

where $Q_\Lambda(\mathbf{x})$ and $\mathbf{x}_e$ are the quantization and quantization error (2.12)*, respectively, induced by the partition $\mathcal{P}$. We shall call this a* modulo-lattice *operation – and use the notation* $\mathbf{x} \bmod \Lambda$*, or* $\mathbf{x}/\Lambda$ *– whenever there is no ambiguity about the assumed partition of* Λ.

Proposition 2.3.1 (Modulo laws) *The modulo-lattice operation satisfies the* shift-invariance *property*

$$(\mathbf{x} + \lambda) \bmod \Lambda = \mathbf{x} \bmod \Lambda, \quad \forall \lambda \in \Lambda, \tag{2.22a}$$

and the distributive *law,*

$$(\mathbf{x} \bmod \Lambda + \mathbf{y}) \bmod \Lambda = (\mathbf{x} + \mathbf{y}) \bmod \Lambda. \tag{2.22b}$$

Proof If $\mathbf{x} = \lambda' + \mathbf{x}_e$ (with $\lambda' \in \Lambda$ and $\mathbf{x}_e \in \mathcal{P}_0$) is the unique decomposition (2.12) of $\mathbf{x}$ with respect to a partition $\mathcal{P}$, then $\mathbf{x} + \lambda = (\lambda + \lambda') + \mathbf{x}_e$ must be the unique decomposition of $\mathbf{x} + \lambda$ with respect to $\mathcal{P}$, i.e., both $\mathbf{x}$ and $\mathbf{x} + \lambda$ have the same quantization error $\mathbf{x}_e$, which proves the shift-invariance property. The distributive law now follows because the inner modulo operation in (2.22b) amounts to shifting