THOMAS ESPITAU

ALGEBRAIC LATTICES: ALGORITHMIC ASPECTS

ALGEBRAIC LATTICES: ALGORITHMIC ASPECTS

THOMAS ESPITAU

An algorithmic view on the geometry of numbers in number fields and applications to number theory and cryptography

To get the degree of Doctor
Laboratoire d'Informatique de Paris 6
Sorbonne Université

2019

PUBLICATIONS

This manuscript is based on the following articles:

- **Certified lattice reduction** *with Antoine Joux.* Advances in Mathematics of Communications,14,1.
- **Optimal lattice reduction and beyond** *with Paul Kirchner and Pierre-Alain Fouque.* Submitted.
- **Computing generator in cyclotomic integer rings** *with Paul Kirchner and Pierre-Alain Fouque and Alexandre Gélin.* Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques EUROCRYPT 2017
- **Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers** *with Pierre-Alain Fouque, Benoit Gérard, Mehdi Tibouchi.* Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.

Besides the topics covered in this manuscript, we worked on various other topics. Follows here a complete list of publications.

CRYPTANALYTICAL WORK

- **LWE without modular reduction and improved side-channel attacks against BLISS** *with Jonathan Bootle, Claire Delaplace, ierre-Alain Fouque and Mehdi Tibouchi.* Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security ASIACRYPT 2018. In this paper, we present a systematic analysis of the LWE probed without moduli. In particular we demonstrate that the least-square method (or alternatively Babai's nearest plane algorithm) is optimal up-to logarithmic factor in the number of samples needed to retrieve the secret. We then applies this analysis to perform a full key recovery from the leakage of the rejection sampling of the BLISS signature scheme.
- **Loop-abort faults on lattice-based signature schemes and key exchange protocols** *with Pierre-Alain Fouque, Benoit Gérard, Mehdi Tibouchi.* IEEE Transactions on Computers, 67, 11. This work is a refinement and a generalization to key exchange protocols of the algorithms presented in the paper **Loop-abort faults on lattice-based Fiat-Shamir and hash-and-sign signatures**.

- **Loop-abort faults on lattice-based Fiat-Shamir and hash-and-sign signatures** *with Pierre-Alain Fouque, Benoit Gérard, Mehdi Tibouchi.* Proceedings of the International Conference on Selected Areas in Cryptography SAC 2016. This work presents fault attacks on lattice-based signatures built on the two main constructions, Fiat-Shamir and hash-and-sign. The exploitation of these attacks yield a full key recovery with only a few or even a single faulty signature, using a novel dimension reduction technique.

- **Higher-order differential meet-in-the-middle preimage attacks on SHA-1 and BLAKE**. *with Pierre-Alain Fouque and Pierre Karpman* Proceedings of the Annual Cryptology Conference CRYPTO 2015. In this paper, we extend the framework of Knellwolf and Khovratovich on meet-in-the-middle attacks to encompass the high-order differentials à la Knudsen. This allows to mount the best (up-to this date) preimage attacks on SHA-1 and BLAKE hash functions.

CRYPTOGRAPHICAL CONSTRUCTIONS

- **GALACTICS: Gaussian Sampling for Lattice-Based Constant-Time Implementation of Cryptographic Signatures, Revisited** *with Gilles Barthe, Sonia Belaïd, Pierre-Alain Fouque, Mélissa Rossi and Mehdi Tibouchi.* Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security CCS 2019. In this work, we present an attack on the rejection sampling of the BLISS-signature scheme using a variation of the Wirtinger flow phase recovery method in statistics. This attack demonstrates the usefulness of a sound constant time implementation of this scheme. The crux of the construction lies in an approximation of transcendental functions by polynomials, found by lattice reduction over polynomials rings endowed with the 2-Sobolev metric.

- **Masking the GLP lattice-based signature scheme at any order** *with Gilles Barthe, Sonia Belaïd, Pierre-Alain Fouque, Benjamin Grégoire, Mélissa Rossi and Mehdi Tibouchi.* Annual International Conference on the Theory and Applications of Cryptographic Techniques EUROCRPYT 2018 In this paper, we describe the first masked implementation of the GLP lattice signature scheme. We show how to provably mask it in the Ishai–Sahai–Wagner model at any order in a relatively efficient manner, using extensions of the techniques of Coron et al. for converting between arithmetic and Boolean masking. Our proof relies on a generalization of probing security that supports the notion of public outputs.

VERIFICATION OF PROBABILISTIC PROGRAMS

- **An Assertion-Based Program Logic for Probabilistic Programs** *with Gilles Barthe, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and*

Pierre-Yves Strub. European Symposium on Programming ESOP 2018. In this paper, we introduce a sound and relatively complete assertion-based program logic, and demonstrate its expressivity by verifying several classical examples of randomized algorithms using an implementation in the EasyCrypt proof assistant. We also show that this system allows convenient reasoning about complex probabilistic concepts by developing a new program logic for probabilistic independence and distribution law, and then smoothly embedding it into our inital logic. This work demonstrates that the assertion-based approach is not fundamentally limited and suggests that some notions are potentially easier to reason about in assertion-based systems.

- **Proving expected sensitivity of probabilistic programs** *with Gilles Barthe, Benjamin Grégoire, Justin Hsu, Pierre-Yves Strub*. Proceedings of the ACM on Programming Languages POPL 2018. We propose in this paper an average notion of program sensitivity for probabilistic programs—expected sensitivity—that averages a distance function over a probabilistic coupling of two output distributions from two similar inputs. By varying the distance, expected sensitivity recovers useful notions of probabilistic function sensitivity, including stability of machine learning algorithms and convergence of Markov chains.

- ***-Liftings for Differential Privacy**. *with Gilles Barthe, Justin Hsu and Tetsuya Sato* Proceedings of the 44th International Colloquium on Automata, Languages, and Programming ICALP 2017. This work unifies all known notions of approximate lifting, giving cleaner properties, more general constructions, and more precise composition theorems for both styles of lifting, enabling richer proofs of differential privacy. We also clarify the relation between existing definitions of approximate lifting, and generalize our constructions to approximate liftings based on f-divergences.

- **Proving uniformity and independence by self-composition and coupling** *with Gilles Barthe, Benjamin Grégoire, Justin Hsu, Pierre-Yves Strub*. Proceedings of the International Conferences on Logic for Programming, Artificial Intelligence and Reasoning LPAR 17. We demonstrate in this paper that probabilistic couplings can be used for verifying non-relational probabilistic properties. Specifically, we show that the program logic pRHL— whose proofs are formal versions of proofs by coupling—can be used for formalizing uniformity and probabilistic independence.

- **Synthesizing probabilistic invariants via Doob's decomposition** *with Gilles Barthe, Luis María Ferrer Fioriti and Justin Hsu*. Proceedings of the International Conference on Computer Aided Verification CAV 16. We propose a novel procedure to synthesize martingale expressions from an arbitrary initial expression. Contrary to state-of-the-art approaches, we do not rely on constraint solving. Instead, we use a symbolic construction based on Doob's decomposition. This procedure can produce very complex martingales, expressed in terms of conditional expectations.

- **Relational reasoning via probabilistic coupling** *with Gilles Barthe, Benjamin Grégoire, Justin Hsu, Léo Stefanesco and Pierre-Yves Strub.* Proceedings of the conference on Logic for Programming, Artificial Intelligence, and Reasoning LPAR 15. This work aims at giving a systematic account of the use of coupling of random variables for the proof of probabilistic programs. In particular, while the mathematical definition of coupling looks rather complex and cumbersome to manipulate, we show that the relational program logic pRHL—the logic underlying the EasyCrypt cryptographic proof assistant—already internalizes a generalization of probabilistic coupling. With this insight, constructing couplings is no harder than constructing logical proofs. We demonstrate how to express and verify classic examples of couplings in pRHL, and we mechanically verify several couplings in EasyCrypt.

CONTENTS

ACRONYMS

GCD: : Greatest Common divisor

GSO: : Gram-Schmidt Orthogonalization

HNF: : Hermite Normal Form

LLL: : Lenstra-Lenstra-Lovász algorithm

BKZ: : Block Korkine Zolotareff algorithm

DBKZ: : Self Dual Block Korkine Zolotareff algorithm of Micciancio and Walter

L^2 : : Quadratic Lenstra-Lenstra-Lovász algorithm of Nguyen and Stehlé

ADAPTIVE-LLL : : Adaptive precision Lenstra-Lenstra-Lovász algorithm

PIP : : Principal Ideal Problem

SIS : : Short integer solution problem

LWE : : Learning With Error problem

SVP: : Shortest vector problem

CVP: : Closest vector problem

FHE: : Fully Homomorphic Encryption

INTRODUCTION

1 GÉOMÉTRIE DES NOMBRES ET RÉSEAUX

Hermann Minkowski, dans son traité *Geometrie der Zahlen* — la *Géométrie des nombres* en français — ([122]), eut la fertile intuition qu'une géométrisation de la théorie des nombres permettrait de prouver de manière quasi visuelle des résultats abstraits. Cette *Géométrie des Nombres* a en particulier permis de comprendre et de simplifier les résultats portant sur les unités de corps de nombres, ainsi que d'étendre grandement les résultats d'approximation Diophantienne. Un exemple simple mais particulièrement frappant de cette nouvelle manière de « voir »la théorie des nombres est le célèbre *théorème de deux carrés de Fermat*, que l'on peut énoncer ainsi :

Théorème 1.1. *Un nombre premier impair p peut être écrit comme une somme de deux carrés : $x^2 + y^2$, avec $x, y \in \mathbf{Z}$ si et seulement si $p \equiv 1 \mod (4)$.*

Nous en donnerons une preuve quasi intuitive après avoir introduit plus formellement la notion de réseau, par opposition avec la preuve plus *classique*, utilisant uniquement des raisonnements arithmétiques sur les congruences modulaires. Depuis les travaux de Minkowski, les liens entre géométrie et théorie des nombres se sont multipliés et densifiés, élargissant ainsi l'éventail des méthodes pour attaquer des problèmes *a priori* purement arithmétiques.

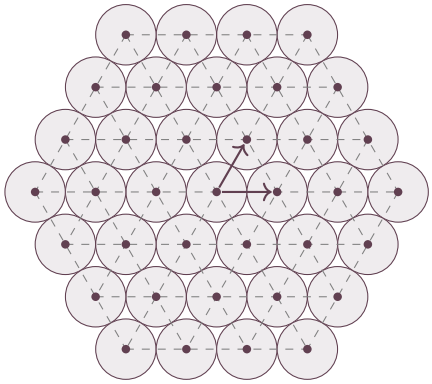
L'objet central de cette théorie géométrique des nombres est le *réseau*, qui formalise l'idée intuitive de *grille* dans le plan ou l'espace. Au travers de cette thèse nous allons nous intéresser à une généralisation de la notion de réseau, non plus dans l'espace Euclidien usuel mais dans des espaces construits sur des corps de nombres, généralisant les nombres rationnels. C'est ce lien étroit entre la théorie des nombres, par essence arithmétique et algébrique, et la géométrie que nous avons souhaité mettre en avant dans ce manuscrit, au travers des méthodes de réduction algorithmique, permettant une approche calculatoire de la théorie développée par Minkowski.

Afin de nous forger une intuition sur cet objet géométrique qu'est le réseau Euclidien, quittons l'arithmétique pour revenir aux prémices de cette notion, qui trouvent leur origine dans un problème au caractère très visuel.

1.1 Genèse : les empilements denses de sphères

Il s'agit du problème dit de « l'empilement de sphères » : quelle est la densité maximale possible d'un empilement de sphères de $\mathbf{R}^n$ pour une dimension n fixée ? Par empilement de sphères de $\mathbf{R}^n$, nous comprenons une famille infinie de boules de même rayon r et d'intérieurs disjoints, et par densité, la fraction de $\mathbf{R}^n$ recouverte par les boules.

En dimension $n = 2$ par exemple, la densité maximale est obtenue grâce à l'empilement dit « hexagonal », où les boules sont centrées sur une grille hexagonale, comme représenté ci-dessous.

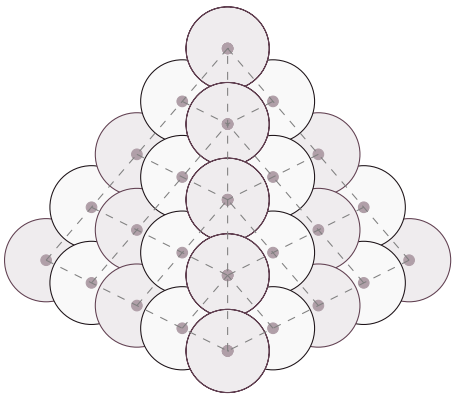


On peut alors montrer que la densité de cet empilement est de :

$$\frac{\pi}{2\sqrt{3}} \cong 0.9069,$$

en comparant l'aire de la surface définie par quatre cercles en contact avec celui du losange définit par les centres de ces cercles. Bien que le caractère optimal de cet empilement soit conjecturé depuis l'école pythagoricienne, il fallut attendre les travaux de Lagrange en 1773 pour en obtenir une preuve rigoureuse.

Le cas de la dimension 3 apparaît comme plus complexe encore. Kepler conjectura en 1611 que l'empilement traditionnel des marchands de fruits, ou celui des canonniers, était le plus dense. Il consiste à empiler les boulets en les laissant glisser dans l'espace formé par ceux de l'étage du dessous, formant une pyramide, comme représenté ci-dessous :



Mais il fallut attendre près de quatre siècles pour obtenir la première démonstration correcte de cette conjecture de Kepler : la preuve ne fut annoncée qu'en 1998 par Hales, et finalement publiée complètement en 2005 et 2006 dans [72, 73]. L'article original de Hales fait près de 300 pages et nécessite une grande quantité de calculs assistés par ordinateurs. La vérification et certification de ces calculs utilise en particulier *l'arithmétique d'intervalles* afin de détecter et de gérer les erreurs d'arrondis des calculs numériques. Nous rencontrerons de nouveau cette technique de calcul, dans le cadre de nos travaux cette fois, pour assurer la certification des algorithmes de réduction de réseaux.

Dans ces deux cas, il est intéressant de remarquer que les centres des boules sont disposés de manière *régulière* dans l'espace : ils forment ce que l'on appelle un *réseau*.

Les cas des dimensions supérieures est resté ouvert jusqu'aux travaux de Viazovska en 2016 puis 2017 — avec ses collaborateurs Cohn, Kumar, Miller et Radchenko cette fois — qui prouvent que les empilements les plus denses en dimensions 8 et 24 proviennent eux aussi du placement de boules sur des réseaux : le réseau E_8 en dimension 8 et le réseau dit de Leech en dimension 24. Il est toutefois amusant de constater que les preuves sont beaucoup plus concises que la preuve de Hales et surtout ne sont pas des preuves purement géométriques : elles reposent au contraire sur l'étude de certaines formes modulaires. Dualement aux idées de Minkowski, c'est cette fois la théorie des nombres qui permet de prouver des théorèmes de géométrie Euclidienne.

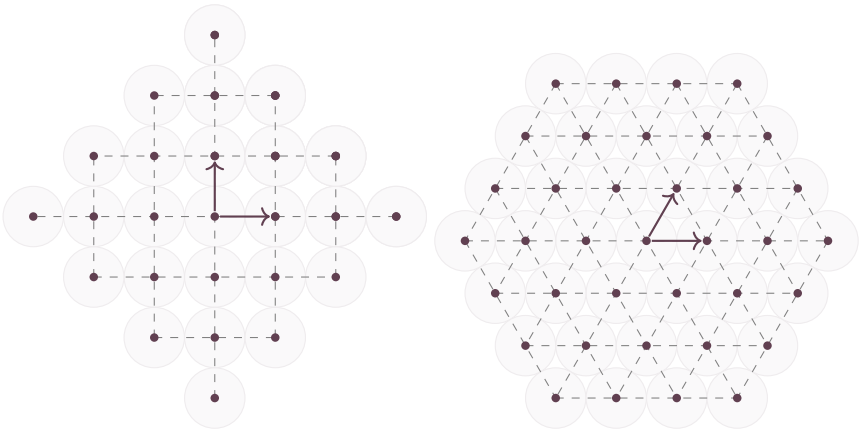
Ces cas très particuliers en dimension 2, 3, 8 et 24 incitent à l'étude des empilements de sphères centrées sur les points d'un réseau, plus simplement nommés «empilements de réseaux». C'est en effet l'une des motivations historiques de l'étude systématique des réseaux Euclidiens. De fait, déterminer la densité de l'empilement de réseau le plus dense revient à déterminer la valeur exacte de la constante dite de «Hermite», que nous introduirons formellement au paragraphe 1.2.3. Si de manière générale les empilements les plus denses sont encore inconnus[1], les empilements de réseaux de petites dimensions ont été, eux, complètement classifiés : par Lagrange en dimension deux [102], Gauss en dimension trois [60], Korkine et Zolotareff en dimensions quatre et cinq [100], Blichfeldt en dimensions six à huit [18], et enfin Cohn et Kumar en dimension vingt-quatre [36]. Les techniques développées pour prouver ces premiers résultats sont des prototypes d'algorithmes que l'on nomme aujourd'hui « réduction de réseaux».

1.2 Réseaux Euclidiens

1.2.1. *Formalisation de la notion de réseau.* Nous avons vu au travers des exemples d'empilements de sphères qu'un réseau est une structure géométrique vivant dans l'espace Euclidien $\mathbf{R}^n$, et formant une grille régulière.

1 Si les empilements les plus denses connus sont effectivement des empilements de réseaux, la généralisation de ce phénomène ne semble pourtant pas aller de soi : cela constitue un problème actuellement ouvert.

Par grille nous entendons que les vecteurs du réseau *pavent* régulièrement l'espace. Afin d'obtenir une représentation visuelle, reprenons le réseau des centres formé par l'empilement hexagonal et l'empilement carré, dans le plan $\mathbf{R}^2$:



Puisqu'il est possible de translater une telle grille sans changer ses propriétés géométriques, nous pouvons toujours supposer que l'origine (0) de l'espace est elle aussi dans le réseau. Dès lors, la somme de deux vecteurs du réseau est aussi un vecteur du réseau, ainsi que l'opposé de tout vecteur. Ainsi un réseau possède une structure algébrique naturelle de *groupe abélien*. En outre, comme l'indiquait l'exemple des empilements, deux vecteurs du réseau ne peuvent être arbitrairement proches : topologiquement, l'ensemble est *discret* pour la distance Euclidienne. Il s'avère que ces deux notions encodent exactement les propriétés d'un réseau, défini de la manière suivante :

Définition 1.1. *Un réseau Euclidien Λ est un sous-groupe discret de $(\mathbf{R}^n,+)$, ou, de manière équivalente, il s'agit de l'ensemble des combinaisons linéaires à coefficients dans $\mathbf{Z}$ d'une famille de vecteurs linéairement indépendants.*

De manière générale, notons $\Lambda[v_1,\dots,v_k]$ le sous-réseau engendré par une famille $(v_1,\dots,v_k)$ de vecteurs de Λ, c'est-à-dire :

$$\Lambda[v_1,\dots,v_k] = \{a_1v_1 + \cdots + a_kv_k \mid a_1,\dots,a_k \in \mathbf{Z}\}.$$

Un réseau est donc une structure *régulière* — la structure des voisins de chaque point étant la même pour tout point — et *discrète* d'un espace, qui est munie d'une norme Euclidienne.

1.2.2. *Bases et covolume.* Une base d'un réseau Λ est une famille $(v_1,\dots,v_k)$ de vecteurs linéairement indépendants tels que ces vecteurs génèrent le réseau, c'est-à-dire telle que :

$$\Lambda[v_1,\dots,v_k] = \Lambda.$$

Manifestement, un réseau de dimension 1 est de la forme $\ell\mathbf{Z} = \{\ell k \mid k \in \mathbf{Z}\}$ pour un certain réel ℓ, et par conséquent n'a que deux bases ℓ et $-\ell$. En revanche, dès que la dimension excède un, un réseau possède une infinité de

bases. Elles ont toutes le même cardinal appelé « rang du réseau ». Les bases diffèrent entre elles par des matrices de passage dites « unimodulaires » : si $B = (b_1, \dots, b_n)$ est une base de Λ, alors une famille $C = (c_1, \dots, c_n)$ de $\mathbf{R}^n$ est aussi une base de Λ si et seulement si la matrice carrée U de taille $n \times n$ exprimant C dans la base B est une matrice inversible — pour conserver la propriété d'être une base — à coefficients entiers — puisque les deux familles sont composées d'éléments du réseau. Il s'avère que ces conditions sur U sont équivalentes au simple fait d'être entière et de déterminant ± 1.

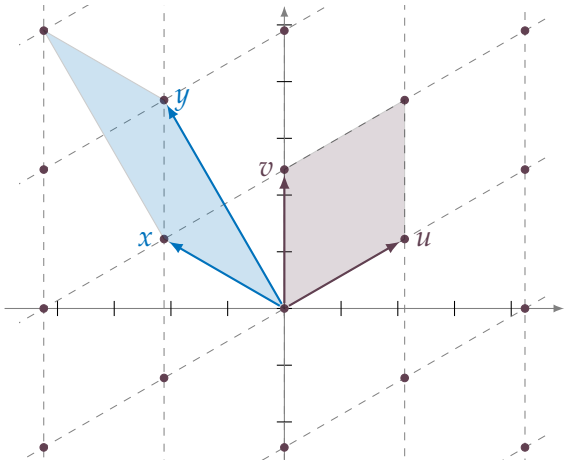


FIGURE 1 : Réseau plan, avec deux bases $B = (u, v)$ et $C = (x, y)$ ainsi que leurs parallélogrammes respectivement associés $\mathcal{P}(B)$ et $\mathcal{P}(C)$. Les aires de ces deux parallélogrammes sont comme annoncé égales.

En particulier, ceci implique que le déterminant $\Delta(b_1, \dots, b_n)$ de la matrice de Gram

$$\mathcal{G} = \left(\langle b_i, b_j \rangle\right)_{1 \leq i,j \leq n}$$

est un réel strictement positif indépendant de la base B choisie : on l'appelle le discriminant du réseau Λ. Le *déterminant* ou *covolume* de Λ, noté covol Λ, est défini quant à lui comme la racine carrée du discriminant. Il est égal au volume — pour la mesure de Lebesgue de $\mathbf{R}^n$— du parallélépipède défini par la base B dans $\mathbf{R}^n$:

$$\mathcal{P}(B) = \left\{ \sum_{i=1}^{d} x_i v_i \,|\, 0 \leq x_i < 1 \right\}.$$

La Figure 1 donne un exemple de réseau plan avec deux bases et leurs parallélépipèdes correspondants.

1.2.3. *Invariant d'Hermite.* Comme nous en avions l'intuition, le caractère discret et périodique d'un réseau assure que ses éléments ne peuvent pas être arbitrairement près les uns des autres et en particulier de l'origine 0. De fait, il est possible de rechercher les plus courts vecteurs du réseau, c'est-à-dire les éléments étant les plus proches de 0. On appelle « premier minimum du réseau » la quantité $\lambda_1(\Lambda) = \min_{x \in \Lambda \setminus \{0\}} \|x\|$.

Les fondements de la géométrie des nombres telle qu'initiée par les travaux de Minkowski permettent de prouver l'existence d'un point du réseau dans des parties de l'espace suffisamment grandes :

Théorème 1.2 (Minkowsi)**.** *Soit Λ un réseau de $\mathbf{R}^n$ et $C \subseteq \mathbf{R}^n$ une partie mesurable, convexe, symétrique par rapport à o, et telle que*

$$Vol(C) > 2^n \operatorname{covol} \Lambda,$$

alors C contient au moins un point de Λ.

L'intuition qui soutient ce théorème est relativement simple : on peut s'en convaincre en juxtaposant les parallélépipèdes formés par les 2^n choix d'orientations possibles des vecteurs d'une base B. L'intérieur de cet ensemble est de volume $2^n \operatorname{vol} \mathcal{P}(B) = 2^n \operatorname{covol} \Lambda$, et par construction, il ne peut contenir que 0. La Figure 2 fournit un exemple en dimension 2 de ce collage.

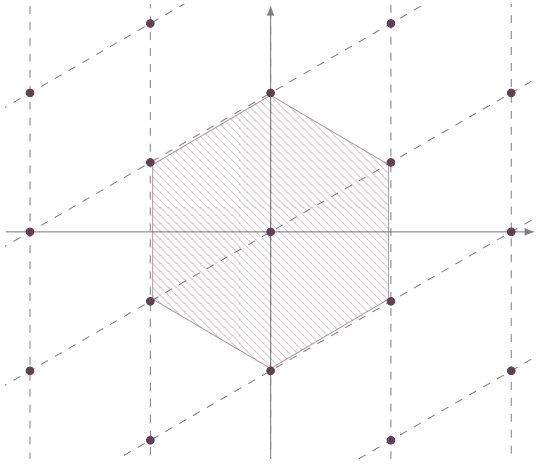


FIGURE 2 : Exemple de cas critique du théorème de Minkowski.

Si l'on applique ce théorème en prenant pour ensemble C des boules fermées, on prouve que tout réseau Λ de dimension n contient un vecteur différent de 0, dont la taille vérifie :

$$\|v\| = 2\left(\frac{\operatorname{covol}(\Lambda)\Gamma\left(\frac{n}{2}+1\right)}{\pi^{\frac{n}{2}}}\right)^{\frac{1}{n}} \leq \frac{1}{2}\sqrt{4+n}(\operatorname{covol}\Lambda)^{\frac{1}{n}},$$

avec Γ la fonction Gamma d'Euler[2]. Le terme de droite de cette équation ne dépend que de la dimension et du volume du réseau Λ. Ainsi, nous pouvons nous demander quel est le meilleur majorant γ_n, tel que pour tout réseau Λ de rang n il existe un vecteur non nul $v \in \Lambda$ tel que $\|v\| \leq \gamma_n(\operatorname{covol}\Lambda)^{\frac{1}{n}}$. Ceci revient donc à évaluer la constante :

$$\sqrt{\gamma_n} = \max_{\Lambda}\left[\frac{\lambda_1(\Lambda)}{(\operatorname{covol}\Lambda)^{\frac{1}{n}}}\right],$$

2 Le volume de la boule de dimension n et de rayon r est en effet égal à $\frac{\pi^{n/2}r^n}{\Gamma\left(\frac{n}{2}+1\right)}$.

où le minimum est pris sur l'ensemble des réseaux réels de rang n. Cette quantité est appelée « la constante de Hermite en dimension n ». Puisque chaque réseau induit un empilement de sphères de rayon $\lambda_1(\Lambda)/2$ et ayant pour centres les points du réseau, il s'avère qu'estimer la valeur de la constante d'Hermite revient à déterminer la densité de l'empilement de réseau le plus dense. La détermination exacte de la constante d'Hermite constitue de fait l'un des problèmes principaux de la géométrie des nombres.

Il est intéressant de noter à ce point que les résultats évoqués permettent d'assurer l'existence *théorique* d'un vecteur court dans un réseau. Toutefois, en pratique, sa construction demeure difficile : la recherche algorithmique du plus court vecteur d'un réseau arbitraire est un problème difficile au sens de la théorie de la complexité[3]. Dès lors, pour des calculs *pratiques*, lorsque la dimension devient trop importante on ne va plus chercher un plus court vecteur mais une « approximation» d'un plus court vecteur, c'est-à-dire un vecteur non nul du réseau se situant dans une boule centrée sur l'origine et de rayon suffisant. Les méthodes algorithmiques permettant cette recherche forment l'algorithmique de la *réduction de réseau*.

1.2.4. *Digression : retour sur la preuve du théorème des deux carrés.* Avant de poursuivre notre tour d'horizon des réseaux et de leur réductions, revenons sur notre théorème liminaire. Nous pouvons en effet donner une preuve concise du théorème des deux carrés de Fermat utilisant la notion de réseau introduite plus haut et en particulier grâce au théorème de Minkowski.

Soit donc p un entier premier congru à -1 modulo 4, ainsi -1 est un résidu quadratique[4] modulo p : il existe donc un entier q tel que $-1 \equiv q^2 \mod p$. Considérons alors le réseau plan Λ engendré par les vecteurs $u = (q, 1)$ et $v = (p, 0)$. Notons M la matrice $[u, v]$. Le volume du réseau est donc par définition :

$$\operatorname{covol} \Lambda = \sqrt{\det(M^T M)} = p$$

Soit alors la partie convexe symétrique par rapport à 0 définie par :

$$C = \left\{(x, y) \in \mathbf{R}^2 \mid x^2 + y^2 < 2p\right\},$$

c'est-à-dire le disque ouvert de rayon $\sqrt{2p}$ centré en 0. Son volume est donc : $\operatorname{vol} C = 2p\pi > 4p = 2^2 p = 2^2 \operatorname{covol} \Lambda$.

Ainsi par le théorème de Minkowski, il existe un point non nul du réseau dans cette boule, (x, y). Mais alors il existe $a, b \in \mathbf{Z}$ tels que : $(x, y) = au + bv = (aq + bp, a)$. Ainsi en prenant le carré de la norme de ce vecteur nous obtenons :

$$x^2 + y^2 = a^2 + a^2q^2 + b^2p^2 + 2abpq \equiv a^2(1 + q^2) \mod p.$$

3 La recherche d'un tel vecteur est en effet un problème NP-difficile, comme montré par Peter van Emde Boas dans [51].

4 En effet, dans le corps fini $\mathbf{F}_p$, -1 est un carré si et seulement si $(-1)^{\frac{p-1}{2}} \neq -1$, c'est à dire si p est congru à 1 modulo 4.

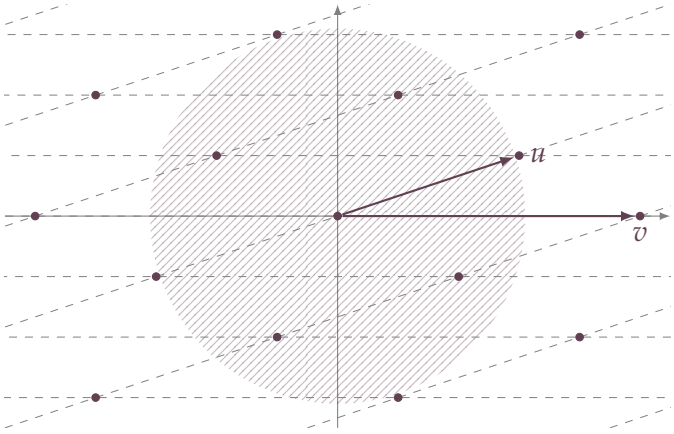


FIGURE 3 : Exemple du réseau obtenu pour $p = 5$, avec le disque C correspondant, hachuré.

Mais comme $q^2 \equiv -1 \mod p$ nous obtenons $x^2 + y^2 \equiv 0 \mod p$, et donc $x^2 + y^2 = p$ puisque $x^2 + y^2 \neq 0$ et $x^2 + y^2 < 2p$. La Figure 3 donne un exemple de la situation pour le cas $p = 5$. On voit clairement apparaître les points à l'intérieur du disque C, permettant de trouver les facteurs carrés comme dans la preuve.

C'est cette dualité entre arithmétique et géométrie qui sera notre fil rouge tout au long de ce manuscrit. Nous avons souhaité développer ce sujet au travers du prisme de l'algorithmique des réseaux, domaine plus récent, qui trouve son origine dans les travaux de Gauss et Lagrange au XIX$^{\text{ème}}$ siècle.

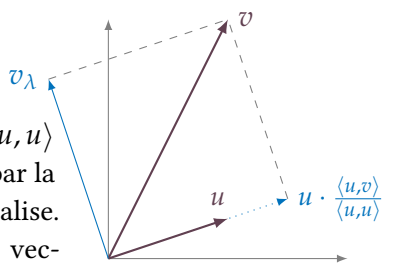
1.3 De la réduction algorithmique des réseaux

Nous avons vu que les réseaux possèdent une infinité de bases dès que la dimension excède deux. Nous savons qu'un espace vectoriel Euclidien possède des bases plus privilégiées : les bases orthonormées. Le caractère continue de $\mathbf{R}^n$ permet en effet de redresser itérativement chaque vecteur d'une base pour assurer son orthogonalité avec les vecteurs précédents.

1.3.1. *Orthogonalisation.* Pour nous en convaincre, regardons comment rendre orthogonal un vecteur v par rapport à un vecteur u, et ce en utilisant seulement une combinaison linéaire de u : soit λ un paramètre réel et $v_\lambda = v - \lambda u$ le vecteur que nous voulons rendre orthogonal à u.

De l'équation $\langle u, v_\lambda \rangle = 0$ nous tirons :

$$\langle u, v \rangle - \lambda \langle u, u \rangle = \langle u, v_\lambda \rangle = 0,$$



de telle sorte que choisir $\lambda = \langle u, v \rangle / \langle u, u \rangle$ convient. Cette construction est illustrée par la figure ci-contre. Cette situation se généralise. Étant donnée une famille $(v_1, \ldots, v_n)$ de vecteurs, nous pouvons commencer par orthogonaliser v_2 par rapport à v_1. No-

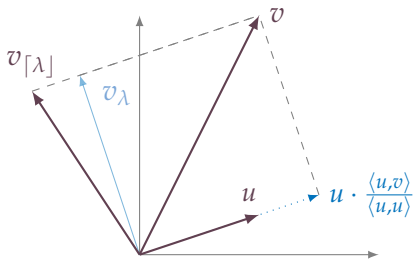
tons v_2^* ce nouveau vecteur. Nous pouvons alors orthogonaliser v_3 par rapport à v_1 et v_2^*, formant un vecteur v_3^* orthogonal à ces deux vecteurs. Nous pouvons continuer avec v_4 en le rendant orthogonal à v_1, v_2^*, v_3^*, et ainsi de suite. Cet algorithme permet de construire une base orthogonale de l'espace telle que le sous-espace engendré par ses i premiers vecteurs soit le même que le sous espace engendré par $v_1, \dots, v_i$. Cette méthode constitue le *procédé d'orthogonalisation de Gram-Schmidt*. Son écriture en pseudo-code est precisée dans l'algorithme 1 :

Algorithme 1 — Gram-Schmidt

Données : $(v_1, \dots, v_n)$ une famille de vecteurs
Résultat : Une famille orthogonale $(v_1^*, \dots, v_n^*)$.

$v_1^* \leftarrow v_1$
pour $k = 2$ **jusqu'à** n **faire**
  $v_k^* \leftarrow v_k$
  pour $j = 1$ **jusqu'à** $k-1$ **faire**
    $v_k^* \leftarrow v_k^* - \left(\frac{\langle v_k, v_j^* \rangle}{\|v_j^*\|^2} \right) \cdot v_j^*$
  fin pour
fin pour
Retourner $(v_1^*, \dots, v_n^*)$

1.3.2. *Réduction en taille.* Dans le cas d'un réseau, en revanche, la rigidité imposée par le caractère discret engendre une obstruction à cette construction : puisque seules des combinaisons à coefficients entiers sont possibles, on ne peut que chercher à *s'approcher* des vecteurs résultants de l'orthogonalisation que nous venons d'introduire.



En effet, si nous reprenons le cas exposé précédemment avec les deux vecteurs u et v, nous ne pouvons plus chercher une combinaison linéaire *arbitraire* de u et v mais seulement des combinaisons à coefficients entiers. Ainsi il n'est plus possible de rendre v_λ et u orthogonaux, mais nous pouvons tâcher de minimiser le défaut d'orthogonalité, c'est-à-dire de trouver la valeur entière de λ minimisant la fonction $\lambda \mapsto \langle v_\lambda, u \rangle$, c'est-à-dire l'entier le plus proche du réel $\lambda = \langle u, v \rangle / \langle u, u \rangle$. De la même manière que le procédé de Gram-Schmidt permet d'orthogonaliser des vecteurs, nous pouvons réduire itérativement une famille $(v_1, \dots, v_n)$ de vecteurs d'un réseau en *discrétisant* les opérations à chaque étape. Au lieu de chercher à rendre orthogonal un vecteur par rapport à chacun des précédents, nous nous efforçons à réduire au maximum le défaut d'orthogonalité. Il s'avère que faire diminuer ce défaut permet aussi de diminuer la

taille des vecteurs considérés. Ainsi ce processus porte le nom de «réduction en taille» ou de «réduction faible». Cet algorithme est décrit en pseudo-code dans l'algorithme 2 .

Algorithme 2 — Réduction en taille

Données : $(v_1, \dots, v_n)$ une famille de vecteurs
Résultat : Une famille réduite en taille $(v_1, \dots, v_n)$.

$v_1^*, \dots, v_n^* \leftarrow$ **Gram-Schmidt**$(v_1, \dots, v_n)$
pour $k = 2$ **jusqu'à** n **faire**
  pour $j = i - 1$ **jusqu'à** 1 **faire**
    $v_k \leftarrow v_k - \left\lceil \frac{\langle v_k, v_j^* \rangle}{\|v_j^*\|^2} \right\rfloor \cdot v_j$
  fin pour
fin pour
Retourner $(v_1, \dots, v_n)$

1.3.3. *Densifier les réseaux successifs.* La notion de base privilégiée semble donc moins immédiate dans le cadre d'un réseau que dans un espace Euclidien. Comme nous l'avons vu, la réduction en taille permet de réduire le défaut d'orthogonalité et la norme des vecteurs d'une base. Nous pouvons donc chercher une base formée de vecteurs aussi courts que possible. Pour ce faire nous pouvons par exemple nous efforcer de *densifier* les sous-réseaux successifs engendrés par les vecteurs de la base, c'est-à-dire d'en faire diminuer le volume. Plus spécifiquement, prenons une base $(v_1, \dots, v_n)$ du réseau Λ, et dénotons par Λ_i le sous-réseau engendré par les i premiers vecteurs $v_1, \dots, v_i$. Supposons que Λ_1 soit le sous-réseau le plus dense parmi tous les sous-réseaux de rang 1. En conséquence, v_1 est nécessairement le vecteur le plus court de Λ. Ensuite, si Λ_2 est lui aussi dense et contient v_1, alors v_2 sera à son tour vraisemblablement court (bien que pas né'essairement le plus court), sans quoi le volume de ce sous-réseau serait grand. Il en va de même pour les sous-réseaux Λ_i suivants : avoir des sous-réseaux Λ_i denses implique que les vecteurs v_i sont petits.

1.3.4. *Vers la réduction LLL.* Il est donc manifeste que la réduction en taille agit sur les vecteurs sans modifier pour autant les sous-réseaux Λ_i. Afin de pouvoir pouvoir densifier ces sous-réseaux successifs, nous pouvons *permuter* les vecteurs de la base. Il serait bien entendu très coûteux d'essayer les $n!$ permutations des n vecteurs. Mais rappelons-nous que le groupe des permutations est engendré par les transpositions de la forme $(i, i+1)$— c'est-à-dire les transpositions de deux éléments adjacents. De fait, nous pouvons chercher à densifier les sous-réseaux successifs en transposant deux à deux des vecteurs successifs. Une fois que des vecteurs ont été échangés, il devient possible d'effectuer une nouvelle réduction en taille. Cette procédure peut alors continuer jusqu'à ce que plus aucune modification, échange, ou réduc-

tion en taille, n'agisse sur la base. Elle sera de fait *réduite*. Ce processus se transcrit alors aisément en pseudo-code :

1. **Tant qu'** une modification est possible **faire**
2. Réduire en taille la base
3. **Si** il existe un indice $1 \leq i \leq n$ tel que le volume du sous-réseau engendré par $v_1, \ldots, v_{i-1}, v_{i+1}$ soit plus petit que le volume du sous-réseau engendré par $v_1, \ldots, v_{i-1}, v_i$ **alors** échanger les vecteurs v_i et v_{i+1}.
4. **fin faire**

En gardant en mémoire le lieu du dernier échange, nous pouvons éviter de repartir au début de la base à chaque itération. En outre nous introduisons une relaxation de la condition d'échange par un paramètre $0 < \delta < 1$. Cette modification permet en réalité de garantir un temps de calcul *polynomial*. En tout et pour tout nous obtenons l'algorithme 3. Cet algorithme est exactement[5] l'algorithme de réduction de *Lenstra-Lenstra-Lovàsz* (LLL), introduit en 1982 et qui a pour application la factorisation de polynômes entiers. Une base sera dite LLL-réduite si elle satisfait les deux propriétés suivantes :

1. Elle est réduite en taille.
2. Aucune transposition de vecteur ne permet d'améliorer la densité au des sous-réseaux successifs $\Lambda[v_1], \Lambda[v_1, v_2], \ldots, \Lambda$.

De fait, on vérifie que l'algorithme LLL renvoie bien une base LLL-réduite d'un réseau à partir d'une base arbitraire donnée en entrée. Si l'on note B une borne sur le nombre de bit nécessaire pour stocker chaque coefficient d'une base d'un réseau Λ de rang n, alors, la réduction LLL s'exécute en $O(n^5 B^3)$ operations binaires. Des variantes plus rapides, utilisant l'arithmétique flottante permettent cependant de réduire un tel réseau en seulement $O(n^4 B \log B)$ comme par exemple avec l'algorithme présenté dans [128].

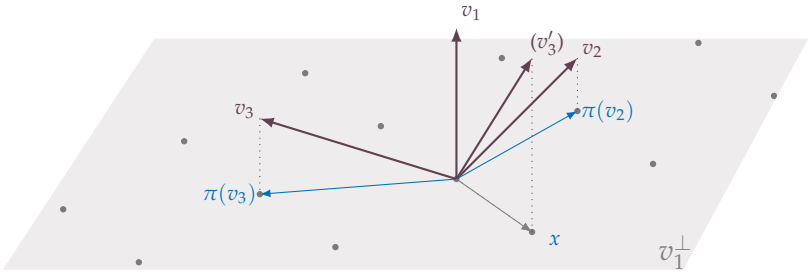
5 Le lecteur familier de cet algorithme remarquera que la condition d'échange par les volumes correspond *exactement* à la condition dite de Lovàsz sur la norme des vecteurs projetés.

Algorithme 3 — LLL-réduction

Données : $(v_1, \dots, v_n)$ une base d'un réseau Λ de rang d
Résultat : Une base réduite de Λ

$k \leftarrow 2$
tant que $k \leq d$ **faire**
  $v_1^*, \dots, v_n^* \leftarrow$ **Gram-Schmidt**$(v_1, \dots, v_n)$
  pour $j = k-1$ **jusqu'à** 1 **faire**
    $v_k \leftarrow v_k - \left\lceil \frac{\langle v_k, v_j^* \rangle}{\|v_j^*\|^2} \right\rfloor \cdot v_j$
  fin pour
  si $\mathrm{vol}(v_1, \dots, v_{i-1}, v_{i+1}) < \delta \,\mathrm{vol}(v_1, \dots, v_{i-1}, v_i)$ **alors**
    $k \leftarrow k+1$
  sinon
    échanger v_k et v_{k-1}
    $k \leftarrow \max(k-1, 2)$
fin tq
Retourner $(v_1, \dots, v_n)$

Le schéma suivant illustre une étape de l'algorithme LLL en dimension 3. La base en cours de réduction est v_1, v_2 , v_3. On cherche à réduire v_3 par le vecteur v_2. Cette étape va s'effectuer, en *relevant* le résultat, noté x de la réduction de $\pi(v_3)$ par $\pi(v_2)$, où π est la projection orthogonale sur l'espace $v_1^\perp$. Ce relèvement est effectué en choisissant l'élément v_3' de $\Lambda \cap (x + v_1)$ de plus petite norme.



1.3.5. *Facteur d'approximation et réduction avec oracles.* L'algorithme LLL est un algorithme de type *glouton* puisqu'il provoque un échange de deux vecteurs dès qu'une amélioration de volume est possible. Néanmoins il n'est pas optimal vis-à-vis de la longueur du premier vecteur ou de la densité des sous-réseaux successifs retournés. En revanche il fournit une *approximation* des valeurs optimales de ces volumes, quantifiée par le lemme suivant :

Lemme 1.1. *Soit $(v_1, \dots, v_n)$ une base d'un réseau Λ, qui est réduite par l'algorithme LLL de paramètre $0 < \delta < 1$. Alors, pour tout $1 \leq k \leq n$, nous avons :*

$$\mathrm{covol}\,\Lambda[v_1, \dots, v_k] \leq \left(\delta - \frac{1}{4}\right)^{-\frac{(n-k)k}{4}} \mathrm{covol}\,\Lambda^{\frac{k}{n}}$$

Par exemple, ce lemme assure que l'algorithme LLL permet de trouver un vecteur de norme plus petite que $\left(\delta - \frac{1}{4}\right)^{-\frac{n-1}{4}} \operatorname{covol} \Lambda^{\frac{1}{n}}$. Sachant que la constante d'Hermite γ_n est polynomiale en n, il s'agit donc d'une approximation exponentielle du plus court vecteur de Λ. Nous pouvons naturellement nous demander s'il est possible de diminuer les termes exponentiels en $(\delta - 1/4)^{-k(n-k)/4}$. Mais alors, à quel coût ? En effet ces facteurs, dits *d'approximation*, peuvent être réduits, si l'on accepte de recourir à l'usage d'un oracle permettant de déterminer *exactement* un plus court vecteur dans un réseau arbitraire. L'algorithm dit de semi-reduction, introduit par Schnorr [145] en 1987, peut être vu comme une extension de LLL dans laquelle on ne cherche plus un plus court vecteur dans un réseau projeté de rang 2, mais cette fois dans un réseau projeté de rang β, pour un paramètre $1 < \beta < n$. L'algorithme ainsi obtenu renvoie certes des vecteurs plus courts que LLL, mais sa complexité est exponentielle en β. Il y a donc un compromis entre la qualité de réduction, essentiellement donnée par la taille du plus court vecteur de la base, et le temps nécessaire à ce calcul. L'illustration même de ce compromis temps/qualité est donnée par le théorème suivant, qui provient de la variante DBKZ de Micciancio et Walter [121], avec les techniques d'énumération de [11] pour l'oracle de recherche du plus court vecteur :

Théorème 1.3. *Le plus petit vecteur v renvoyé par l'algorithme DBKZ avec paramètre β vérifie :*

$$\|v\| \leq \beta^{\frac{n-1}{2(\beta-1)}} \cdot (\operatorname{vol} \Lambda)^{\frac{1}{n}}.$$

Cet algorithme a une complexité de la forme $\mathrm{Poly}(n, \log B)\left(\frac{3}{2}\right)^{\beta/2+o(\beta)}$, où B est une borne sur la taille des coefficients de la base d'entrée.

2 LE CADRE ALGÉBRIQUE : RÉSEAUX ET CORPS DE NOMBRES

2.1 Corps de nombres, nombres algébriques

Le concept de nombre algébrique, introduit par Abel, est né de la volonté des mathématiciens de résoudre les équations « algébriques »c'est-à-dire de résoudre des équations du type $P(x) = 0$, où P désigne un polynôme à coefficients rationnels. L'idée fondamentale, évoquée dans les lettres d'Évariste Galois, consiste à adjoindre une racine α de P aux rationnels $\mathbf{Q}$ et à étudier l'objet

$$\mathbf{Q}(\alpha) = \{q_0 + q_1\alpha + \cdots + q_k\alpha^k \mid k \leq 0, q_1, \ldots, q_k \in \mathbf{Q}\}$$

résultant de cette adjonction. Donnons un exemple de corps de nombres qui nous servira à exemplifier les notions que nous introduirons ultérieurement. On construit ainsi le corps $\mathbf{Q}(i) = \{a + ib \mid a, b \in \mathbf{Q}\}$ provenant de l'adjonction des racines du polynome $X^2 + 1$ au corps des rationnels.

2.1.1. *Structure de corps de nombre.* Il s'avère que $\mathbf{Q}(\alpha)$ est un corps et constitue donc à ce titre une extension des rationnels. Il est en outre de degré

fini — sa dimension $[\mathbf{Q}(\alpha) : \mathbf{Q}]$ est finie en tant que $\mathbf{Q}$ espace-vectoriel — sur $\mathbf{Q}$. De ce fait, tout élément de $\mathbf{Q}(\alpha)$ est *algébrique*, c'est-à-dire annulé par un polynôme à coefficients rationnels. Si nous nous limitons à regarder les éléments de $\mathbf{Q}(\alpha)$ qui sont racines de polynômes *unitaires* à coefficients *entiers* nous obtenons un sous-ensemble $\mathscr{O}_{\mathbf{Q}(\alpha)}$, qui est un anneau pour les lois de $\mathbf{Q}(\alpha)$. Il s'agit de *l'anneau des entiers du corps* $\mathbf{Q}(\alpha)$, qui constitue une généralisation des entiers relatifs — l'anneau des entiers de $\mathbf{Q}$ n'est autre que $\mathbf{Z}$. Il constitue même une généralisation n-dimensionnelle de $\mathbf{Z}$ puisque l'on peut montrer (cf. Theorem 1.2.2) qu'il existe des éléments $v_1, \dots, v_n \in \mathscr{O}_{\mathbf{K}}^n$ tels que $\mathscr{O}_{\mathbf{K}}$ soit l'ensemble des combinaisons linéaires à coefficients dans $\mathbf{Z}$ des v_i, c'est-à-dire que :

$$\mathscr{O}_{\mathbf{K}} \cong v_1\mathbf{Z} \oplus \cdots \oplus v_n\mathbf{Z} \cong \mathbf{Z}^n.$$

Si l'on reprend notre exemple, l'anneau des entiers de $\mathbf{Q}(i)$ est $\mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$ appelé l'anneau *des entiers de Gauss*. Il est immédiat que $\mathbf{Z}[i] \cong \mathbf{Z}^2$ en tant que groupe abélien.

2.1.2. *Norme Euclidienne sur un corps de nombre.* Un corps de nombres $\mathbf{K} = \mathbf{Q}[\alpha]$ peut être *plongé*[6], dans le corps des complexes $\mathbf{C}$ de multiples manières. Puisque le corps $\mathbf{K}$ est engendré par l'élément α, tout plongement σ est parfaitement déterminé par l'image de α. Sachant que pour tout polynôme $P \in \mathbf{Q}[X]$ nous avons nécessairement $\sigma(P(\alpha)) = P(\sigma(\alpha))$, $\sigma(\alpha)$ doit donc être *conjugué* avec α, c'est-à-dire être une racine du polynôme minimal de α. Il y a donc au plus $n = [\mathbf{K} : \mathbf{Q}]$ prolongements possibles et on peut vérifier qu'ils sont tous distincts. Notons les $\sigma_1, \dots, \sigma_n$.

Ces plongements permettent de définir le *plongement Archimédien* du corps $\mathbf{K}$ dans $\mathbf{C}^n$:

$$\sigma : \left| \begin{array}{ccc} \mathbf{K} & \longrightarrow & \mathbf{C}^n \\ x & \longmapsto & (\sigma_1(x), \dots, \sigma_n(x)) \end{array} \right. .$$

Nous pouvons alors *relever* la structure Hermitienne canonique de $\mathbf{C}^n$ vers le corps $\mathbf{K}$ en définissant le produit hermitien :

$$\langle a, b \rangle = \sum_{i=1}^{n} \sigma_i(a)\overline{\sigma_i(b)},$$

où $z \mapsto \bar{z}$ est la conjugaison habituelle de $\mathbf{C}$. Ce produit munit $\mathbf{K}$ de sa *norme canonique*, donnée par $\|a\| = \sqrt{\langle a, a \rangle}$.

Dans notre exemple, il existe donc deux plongements σ_1 et σ_2, envoyant respectivement la racine i sur elle-même et sur sur son conjugué $-i$. Ainsi, le plongement Archimédien d'un élement $a + ib \in \mathbf{Q}$ est le vecteur $(a + ib, a - ib) \in \mathbf{C}^2$. La norme algébrique de $a + ib$ est donc $(a + ib)(a - ib) = a^2 + b^2$.

Mais quel rapport avec les réseaux, introduits *supra* ?

6 Dans le sens présent le plongement est *algébrique*, c'est-à-dire qu'il préserve les propriétés de corps de $\mathbf{K}$: un plongement n'est donc rien d'autre qu'un morphisme de corps de $\mathbf{K}$ dans $\mathbf{C}$.

2.2 Structure naturelle de réseaux des entiers de corps de nombre

Puisque l'anneau des entiers $\mathscr{O}_{\mathbf{K}}$ de $\mathbf{K}$ est inclus dans $\mathbf{K}$, il hérite de la structure Euclidienne que nous venons de définir. Ainsi puisque $\mathscr{O}_{\mathbf{K}}$ est l'ensemble des combinaisons linéaires entières d'une base, il s'agit bien d'un réseau au sens de la définition que nous avons donnée en paragraphe 1.2.1.

2.3 Réseaux algébriques

Nous avons introduit la notion de réseau comme étant un ensemble de combinaisons linéaires à coefficients entiers, vivant dans un espace Euclidien. La structure algébrique décrivant formellement l'ensemble des combinaisons entières d'une base est celle de $\mathbf{Z}$-*module* libre.

2.3.1. *Petite digression sur les modules.* Rappelons brièvement qu'un module est une structure algébrique définie comme un espace vectoriel, mais dans laquelle le corps des scalaires n'est plus un corps mais seulement un anneau. Il s'agit donc d'un groupe additif muni d'une loi de multiplication externe par les éléments d'un anneau, avec les bonnes règles de compatibilité. Cette structure est plus subtile que celle d'espace vectoriel, en particulier car un anneau est beaucoup moins *rigide* qu'un corps et la théorie des modules sur des anneaux arbitraires est très riche. Néanmoins notons qu'un module *libre* finiment généré sur un anneau R se comporte essentiellement comme un espace vectoriel. En particulier le module possède des bases de même cardinal. Il est, de fait, isomorphe à R^n pour un certain n, de la même manière qu'un $\mathbf{K}$-espace vectoriel de dimension n est isomorphe à $\mathbf{K}^n$. Ainsi, un $\mathbf{Z}$-module libre n'est rien d'autre qu'un « espace vectoriel »sur $\mathbf{Z}$, c'est-à-dire une structure de la forme $b_1\mathbf{Z} \oplus \cdots \oplus b_n\mathbf{Z}$ pour $b_1, \ldots, b_n$ une base.

2.3.2. *Généralisation de la notion de réseau.* Nous pouvons par conséquent réinterpréter de la manière suivante la définition de réseau donnée précédemment. Un réseau est la donnée de deux structures : d'une part la structure *algébrique*, à savoir celle de $\mathbf{Z}$-module libre que nous venons de détailler, et d'autre part sa structure *métrique*, c'est-à-dire, celle de norme Euclidienne sur ses éléments. Nous allons maintenant tâcher de généraliser la notion de réseau en remplaçant l'anneau des entiers $\mathbf{Z}$ par des anneaux plus généraux.

Une famille de candidats naturels pour ces anneaux est constituée par les anneaux d'entiers de corps de nombres. Ces anneaux possèdent une arithmétique proche de l'arithmétique usuelle de $\mathbf{Z}$. Ainsi, donnons-nous $\mathbf{K}$, un corps de nombre, et notons $\mathscr{O}_{\mathbf{K}}$ son anneau d'entiers. Métriquement parlant, nous disposons d'une norme Euclidienne sur $\mathbf{K}$, qui permet de construire une norme Euclidienne sur l'espace vectoriel $\mathbf{K}^n$ pour n un entier fixé. Nous n'allons plus considérer des modules sur $\mathbf{Z}$ mais des modules sur $\mathscr{O}_{\mathbf{K}}$. Ainsi

un $\mathcal{O}_{\mathbf{K}}$-réseau de rang n sera un $\mathcal{O}_{\mathbf{K}}$-module[7] de rang n, muni de la norme Euclidienne de $\mathbf{K}^n$.

Les travaux que nous allons présenter dans ce manuscrit visent à étendre l'algorithmique des réseaux Euclidiens aux réseaux sur des corps de nombres.

3 CONTRIBUTIONS

3.1 Du problème de la représentation et de la réduction prouvable

Nous avons vu que nous pouvions construire une métrique adaptée à un réseau sur un corps en utilisant les plongements dans le corps des complexes. Il est utile de remarquer que ces plongements sont calculés comme des évaluations de polynômes sur les racines du polynôme de définition du corps de nombre. Ainsi, une fois les racines approximées, grâce aux raffinements de la méthode itérative de Newton par exemple [69], les calculs de produits scalaires entre vecteurs du réseau algébrique seront également des approximations de leur valeur exacte. Si nous lançons l'algorithme de réduction LLL sur cette version approchée du réseau, nous ne pouvons pas garantir que la base renvoyée soit effectivement LLL-réduite.

3.1.1. *Une arithmétique des intervalles.* L'arithmétique d'intervalles est une méthode de calcul consistant à manipuler des intervalles afin de représenter des nombres (par exemple entiers ou flottants), afin d'obtenir des résultats rigoureux. Cette approche permet de borner les erreurs d'arrondi dans des calculs approchés et ainsi de développer des méthodes numériques certifiantes qui fournissent des résultats fiables.

Si les techniques d'encadrement de nombres réels font partie du folklore commun en calcul numérique — Archimède utilisait en effet déjà de tels outils pour ses premières estimations de la constante π au III$^{\text{e}}$ siècle av. J.-C. — en revanche, la formalisation sous forme de règles de l'arithmétique d'intervalles ne date que des travaux de Young [165] en 1931. Par la suite, un article sur l'algèbre d'intervalles appliqué à l'analyse numérique a été publié par Sunaga en 1958 [158]. La naissance de l'arithmétique d'intervalles moderne est ensuite marquée par le livre *Interval Analysis* de Ramon E. Moore en 1966 [125], qui lui donne ses lettres de noblesse. Son mérite est, à partir d'un principe simple, de donner une méthode générale pour estimer les erreurs de calcul et d'imprécision sur les données d'entrée.

Le principe de cette arithmétique est en effet particulièrement simple : pour représenter un nombre $x \in \mathbf{R}$, on va manipuler l'intervalle $[x^-, x^+] \subset \mathbf{R}$, tel que $x \in [x^-, x^+]$. Par conséquent, on pourra additionner, soustraire, multiplier et diviser des intervalles tout en gardant la certification que le résultat se trouve toujours dans l'intervalle-résultat. L'utilité principale de

7 Pour des raisons de compatibilité que nous éclaircirons au Chapitre 3, nous verrons qu'il est trop restrictif de ne considérer que des modules libres, mais qu'assouplir légèrement cette condition en projectivité suffit.

cette arithmétique va être de pouvoir assurer des comparaisons et de détecter au cours du calcul si trop d'erreurs d'approximation ont été faites pour conclure de manière satisfaisante. Regardons par exemple le cas schématisé ci-dessous :

L'intervalle représentant x assure que $x < 0$ puisque la borne supérieure x^+ l'est aussi et que par définition, $x \leq x^+$. Par contre on ne peut pas conclure quant au fait que $y < z$, puisque $y^+ > z^-$, ce qui implique que les intervalles représentant y et z se chevauchent. On peut en revanche inférer à ce point que les approximations effectuées pour obtenir y et z sont trop grosses pour certifier le résultat global $y < z$.

3.1.2. *Réduction de réseaux approximés avec l'arithmétique d'intervalles.* Les implémentations rapides actuelles de l'algorithme LLL utilisent l'arithmétique flottante en basse précision pour accélérer les calculs. Bien entendu des bornes sur la précision minimale à utiliser sont prouvables, comme par exemple pour la variante L^2 de Nguyen et Stehlé [131]. Ces algorithmes supposent que les réseaux sont donnés exactement, c'est-à-dire que les produits scalaires sont calculables et représentables avec la précision utilisée par l'algorithme de réduction. En revanche s'il est impossible ne serait ce que d'obtenir une approximation des produits scalaires, comme dans le cas algébrique évoqué plus haut, nous ne pouvons nous contenter de telles bornes. Il faut être en mesure de détecter que l'approximation utilisée est insuffisante. Cet obstacle est complètement résolu par l'utilisation de l'arithmétique d'intervalles : si un manque de précision dans le calcul est détecté et que la précision des quantités internes est suffisante, alors nous pouvons conclure que la représentation même des plongements du réseau est insuffisante et ainsi les recalculer plus finement. Cette technique permet donc de calculer de manière certifiée des bases réduites pour des réseaux algébriques, en utilisant uniquement la structure métrique du réseau : la réduction d'un $\mathscr{O}_{\mathbf{K}}$-réseau de rang d sur un corps de nombres $\mathbf{K}$ de degré n passe donc par la réduction d'un réseau (approximé) de rang $d \times n$.

3.1.3. *Un algorithme de réduction certifié grâce à l'arithmétique d'intervalles.* Une conséquence intéressante de cette réduction est qu'elle permet de certifier que tout le calcul de la réduction se fait comme l'aurait fait une version de LLL en arithmétique *exacte*. En particulier, ceci permet l'étude expérimentale de la version exacte de l'algorithme LLL avec la rapidité des implémentations en algorithmique flottante, tout en conservant l'assurance que la trace d'exécution est celle de la version exacte.

La réduction de réseau dans le cas moyen, c'est-à-dire l'étude des algorithmes de réduction lancés sur des réseaux aléatoires[8] est encore très mal

8 Il est possible de définir formellement une notion de réseaux aléatoires en renormalisant la mesure de Haar sur l'espace des modules des réseaux.

comprise. Par exemple, les résultats pratiques de [130] assurent qu'une base LLL réduite l'est bien mieux que la borne théorique du pire cas. En revanche, les résultats théoriques de [95] prouvent que l'immense majorité des bases LLL-réduites sont des bases atteignant les bornes de pire-cas. L'algorithme LLL *choisit* donc naturellement des bases de meilleure qualité que ce que son analyse ne laisse supposer. Il est de fait très avantageux de pouvoir utiliser un algorithme rapide qui possède la même trace d'exécution que la version originale du LLL que l'on souhaite étudier.

3.2 Vers une réduction plus rapide

Grâce à l'arithmétique d'intervalles, il est donc possible d'assurer une notion de réduction prouvée pour des réseaux généraux et en particulier pour les réseaux algébriques. Néanmoins, la réduction des $\mathscr{O}_{\mathbf{K}}$-réseaux commence par faire *descendre* le $\mathscr{O}_{\mathbf{K}}$-réseau sur $\mathbf{Z}$. On oublie la structure algébrique de $\mathscr{O}_{\mathbf{K}}$ et on lance la réduction LLL. Or l'image sur $\mathbf{Z}$ d'un $\mathscr{O}_{\mathbf{K}}$-réseau Λ de rang d est de rang $d \times n$, où n est le degré de $\mathbf{K}$: ainsi, même un réseau de petit rang sur $\mathscr{O}_{\mathbf{K}}$ peut nécessiter une réduction sur $\mathbf{Z}$ en grande dimension, qui est nécessairement coûteuse.

Lors d'un tel procédé, on laisse complètement de côté les spécificités algébriques de l'anneau $\mathscr{O}_{\mathbf{K}}$. Or ces propriétés se traduisent par des symétries sur les $\mathscr{O}_{\mathbf{K}}$ modules : ils sont très structurés. Par conséquent, la réduction évoquée ne tient pas compte des symétries produites par cette structure. Il est donc naturel de se demander s'il n'est pas possible *d'exploiter* la structure algébrique de $\mathscr{O}_{\mathbf{K}}$ pour accélérer la réduction.

3.2.1. *Réduction sur les anneaux Euclidiens.* Une première porte vers une algorithmique générique des réseaux algébriques consiste à étudier la situation sur des anneaux dont l'arithmétique est la plus proche de l'arithmétique des entiers relatifs. Un rapide regard sur la taxonomie classique des anneaux révèle que les anneaux dits « Euclidiens »sont parmi les plus proches des entiers relatifs. Pour de tels anneaux l'algorithme de réduction LLL peut être adapté presque sans modification[9]. Cette variation a été introduite par Napias dans [126], puis améliorée dans [30].

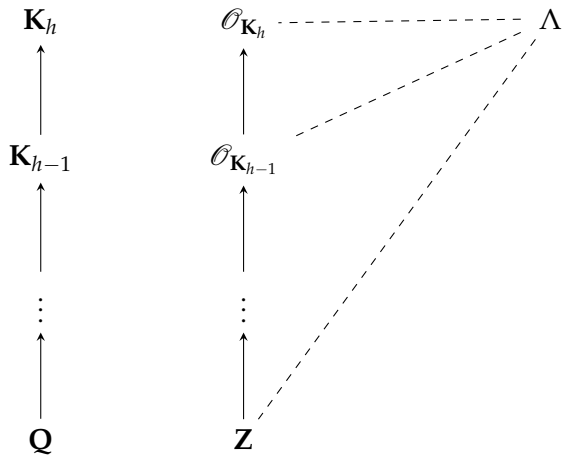
Bien que cette généralisation soit simple et permette effectivement la réduction de réseaux algébriques, nous sommes très loin du cas général puisque les anneaux Euclidiens sont rares[10]. Regardons alors un cas plus général, qui sera étudié en détail au Chapitre 5.

3.2.2. *Récursion sur les tours de corps de nombres.* Considérons un corps de nombres $\mathbf{K}_h$, situé au sommet d'une *tour* de corps de nombres : $\mathbf{Q} \subseteq \mathbf{K}_0 \subseteq \cdots \subseteq \mathbf{K}_{h-1} \subseteq \mathbf{K}_h$, ainsi qu'un $\mathscr{O}_{\mathbf{K}_h}$-réseau Λ.

9 Il suffit en effet de remplacer la valeur absolue sur $\mathbf{Q}$ par la norme algébrique de l'anneau.

10 Parmi les anneaux norme-Euclidiens connus, citons les petits corps quadratiques imaginaires et les petits cyclotomiques[11]

Puisque nous pouvons voir Λ comme réseau sur n'importe lequel des anneaux d'entiers intermédiaires $\mathcal{O}_{\mathbf{K}_i}$, l'existence d'une telle tour de corps permet d'envisager une approche naturellement *récursive* de la réduction. En effet, nous savons réduire tout type de réseau sur $\mathbf{Z}$, anneau présent en bas de la tour, qui constitue *de facto* le cas de base de notre algorithme.



3.2.3. *Structure générale de la réduction.* La structure générale de notre algorithme de réduction récursif est proche de celle de la réduction LLL classique : outre des passes de réduction en taille qui permettent de maîtriser la taille des coefficients lors du calcul, le cœur de l'algorithme consiste en la réduction de réseaux de rang 2, projetés orthogonalement aux vecteurs précédents. Nous conservons ici cette idée, qui autorise l'extension de la réduction pour les réseaux de rang 2 aux réseaux de rang arbitraires.

Il s'agit donc d'être capable de réduire des $\mathcal{O}_{\mathbf{K}_h}$-réseaux de rang 2, Λ, correspondant à la projection orthogonale d'une paire de vecteurs de la base en cours de réduction. Pour ce faire, nous utilisons la structure récursive de la tour pour « voir »le réseau Λ comme $\mathcal{O}_{\mathbf{K}_{h-1}}$—réseau de rang $2 \times [\mathbf{K}_h : \mathbf{K}_{h-1}]$ et appeler la réduction récursivement sur ce nouveau réseau. À chaque appel récursif nous descendons dans la tour, si bien que les feuilles vont correspondre à la réduction de réseaux sur un anneau qui sera Euclidien[12]. Or nous savons réduire directement les réseaux sur un anneau Euclidien. Ainsi en utilisant une telle réduction pour traiter les feuilles de l'arbre de descente, nous pouvons renvoyer un réseau réduit.

Revenons donc au premier appel récursif : lorsque l'appel se termine, nous avons pu réduire notre réseau de rang $2 \times [\mathbf{K}_h : \mathbf{K}_{h-1}]$, correspondant à la descente sur $\mathcal{O}_{\mathbf{K}_{h-1}}$ de la projection orthogonale du $\mathcal{O}_{\mathbf{K}_h}$-réseau b_i, b_{i+1} sur les vecteurs précédents. Cette réduction permet de trouver un vecteur court du plan $\mathcal{P} = b_i\mathcal{O}_{\mathbf{K}_h} \oplus b_{i+1}\mathcal{O}_{\mathbf{K}_h}$. Il suffit alors de construire un second vecteur capable de compléter le premier pour former une nouvelle base de $\mathcal{P}$ afin de continuer la réduction. Cette complétion se fait en résolvant une équation de Bézout à coefficients dans $\mathcal{O}_{\mathbf{K}}$.

12 Rappelons que tout corps de nombres contient $\mathbf{Q}$, si bien que nous sommes certains d'arriver sur $\mathbf{Z}$ par la descente.

3.2.4. *Euclide étendu sur corps de nombres.* Une équation de Bézout est une équation de la forme

$$au + bv = 1$$

d'inconnues u et v dans les entiers relatifs. Elle n'a de solution que si a et b sont *premiers entre eux*. La recherche de solution s'effectue classiquement en temps polynomial par *l'algorithme d'Euclide étendu*. La version généralisée de l'équation de Bézout sur les corps de nombres prend la même forme mais cette fois avec $a, b \in \mathscr{O}_{\mathbf{K}_h}$ des entiers algébriques. Comme en toute généralité l'anneau $\mathscr{O}_{\mathbf{K}}$ n'est pas Euclidien, l'algorithme d'Euclide étendu, basé sur des divisions Euclidiennes successives, ne fonctionne pas en l'état. Néanmoins si la tour $\mathbf{Q} \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h$ n'est pas triviale, nous pouvons nous servir de la structure de cette tour pour descendre le problème sur le sous corps $\mathbf{K}_{h-1}$, en calculant la *norme relative*[13] $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}$ des éléments a et b. Ensuite, en rappelant récursivement l'algorithme de résolution sur $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)$ et $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)$, nous obtenons deux entiers algébriques μ et ν de $\mathscr{O}_{\mathbf{K}_{h-1}}$ solutions de l'équation :

$$\mu N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a) + \nu N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b) = 1. \tag{0.1}$$

En conséquence, nous pouvons prouver que pour tout élément $\alpha \in \mathscr{O}_{\mathbf{K}_h}$, nous avons $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(\alpha) \in \alpha\mathscr{O}_{\mathbf{K}_h}$, de telle sorte que $\alpha^{-1}N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(\alpha) \in \mathscr{O}_{\mathbf{K}_h}$. Alors nous obtenons

$$a \cdot \underbrace{\mu\, a^{-1}N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)}_{:=u\in\mathscr{O}_{\mathbf{K}_h}} + b \cdot \underbrace{\nu\, b^{-1}N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)}_{:=v\in\mathscr{O}_{\mathbf{K}_h}} = 1,$$

comme désiré.

Il suffit alors de noter que les éléments u et v trouvés par cet algorithme ne sont pas nécessairement les plus petites solutions de l'équation de Bézout. Afin d'éviter une explosion de la taille des coefficients nous devons contrôler la taille des solutions qui apparaissent à chaque descente et remontée. Ce contrôle est possible grâce à une technique similaire à celle qui permet de contrôler la taille des coefficients dans l'algorithme de réduction, à savoir une réduction en taille.

Ainsi, la combinaison de cette remontée avec l'algorithme que nous avons évoqué permet une réduction plus rapide sur les corps de nombres. Pour un réseau algébrique de rang 2 sur un corps cyclotomique de degré n, il suffit de $\mathrm{O}\big(n^2 B \log B\big)$ opérations binaires, au lieu de $\mathrm{O}\big(n^4 B \log B\big)$ que nécessiterait l'algorithme rapide de [128].

3.3 Structure symplectique naturelle dans une tour

Génériquement, un espace Euclidien peut être défini comme un expie vectoriel munit d'une forme bilinéaire symétrique (définie positive). En rem-

13 Pour une extension de corps $\mathbf{K}_{h-1} \subset \mathbf{K}_h$, la norme relative $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a) \in \mathbf{K}_{h-1}$ de $a \in \mathbf{K}_h$ est le déterminant de l'application $\mathbf{K}_{h-1}$ linéaire $x \mapsto ax$ sur $\mathbf{K}_h$ vu comme espace vectoriel. Cette forme est multiplicative : pour tout $a, b \in \mathbf{K}_h$, $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(ab) = N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)$.

plaçant cette forme par une forme antisymétrique, nous obtenons un «espace symplectique». Les réseaux plongés dans des espaces symplectiques possèdent des symmétries qui peuvent être exploitées pour accélérer la réduction en taille—plus précisément, la version arrondie entière de l'orthogonalisation de Gram-Schmidt (ou décomposition QR) est remplacé par une version arrondie de la décomposition d'Iwasawa—. Dans ce manuscrit, nous montrons qu'il est possible de définir génériquement une structure symplectique sur un corps de nombre qui soit compatible avec la descente sur ses sous-corps. Ainsi, à toute tour de corps de nombre nous pouvons associer une structure symplectique compatible avec les descentes le long de la tour. De fait lors de la réduction de réseaux algébriques dans de telles structures, nous pouvons diviser par deux le temps de calcul à chaque étage de la tour, donnant lieu à une accélération asymptotique non négligeable au total. Avec cette technique, nous pouvons réduire un réseau alébrique de rang 2 sur un corps cyclotomique de degré n et de conducteur suffisament lisse, donc les coefficients de la représentation matricielle tiennent sur B bits en temps

$$\tilde{\mathrm{O}}\left(n^{2+\frac{\log(1/2+1/2q)}{\log q}} B\right) + n^{\mathrm{O}(\log\log n)},$$

où q est un nombre premier diviant le conducteur du corps. La première colonne de la matrice réduite à ses coefficients uniformément bornés par $2^{\tilde{\mathrm{O}}(n)} (\operatorname{covol} M)^{\frac{1}{2n}}$.

3.4 Une application en théorie algorithmique des nombres : le problème de l'idéal principal

3.4.1. *Idéaux.* Puisque les entiers algébriques $\mathscr{O}_{\mathbf{K}}$ d'un corps de nombres forment un anneau, nous pouvons regarder ses idéaux[14]. En particulier, si les entiers algébriques ne partagent pas la propriété *d'unicité de la factorisation en premiers*[15], les idéaux en jouissent néanmoins. Plus précisément, tout idéal $\mathfrak{a}$ de $\mathscr{O}_{\mathbf{K}}$ se décompose en un produit unique, à l'ordre près, des idéaux premiers. En ce sens, les anneaux d'entiers possèdent une arithmétique des idéaux qui est similaire à celle des entiers rationnels $\mathbf{Z}$. En revanche, il faut remarquer qu'au contraire des idéaux de $\mathbf{Z}$, les idéaux sur les entiers algébriques ne sont plus nécessairement principaux, c'est-à-dire engendrés par un unique élément. Un problème classique de la théorie algorithmique des nombres est soulevé par cette question : étant donné un idéal de l'anneau des entiers $\mathscr{O}_{\mathbf{K}}$, avec la garantie que cet idéal est principal, à quel point est-il difficile de calculer un générateur ? Afin de faciliter la lecture, dans la suite

14 Rappelons qu'un *idéal* $\mathfrak{a}$ est un sous-groupe pour la loi additive et stable sous multiplication par $\mathscr{O}_{\mathbf{K}}$: tel que pour tout entier $x \in \mathscr{O}_{\mathbf{K}}$, $x\mathfrak{a} \subseteq \mathfrak{a}$. Il est dit premier, si la condition suivante est vérifiée : pour tout $a, b \in \mathscr{O}_{\mathbf{K}}$, $ab \in \mathfrak{a}$ implique que $a \in \mathfrak{a}$ ou $b \in \mathfrak{a}$. Sa norme est définie comme le cardinal du quotient $\mathscr{O}_{\mathbf{K}}/\mathfrak{a}$. Il s'agit d'une forme multiplicative donnant une mesure de la taille de l'idéal $\mathfrak{a}$: plus $\mathfrak{a}$ possède une structure proche de $\mathscr{O}_{\mathbf{K}}$, plus elle sera petite.

15 Citons à titre d'exemple la double factorisation $4 = 2 \times 2 = (1 + i\sqrt{3})(1 - i\sqrt{3})$ dans le corps de nombre $\mathbf{Q}[i\sqrt{3}]$.

de ce paragraphe, nous noterons $\langle a \rangle = a\mathcal{O}_{\mathbf{K}} = \{a \cdot x \,|\, x \in \mathcal{O}_{\mathbf{K}}\}$ l'idéal engendré par a dans $\mathcal{O}_{\mathbf{K}}$ pour tout élément a.

3.4.2. *Solution sur les petits premiers.* Supposons que nous sachions résoudre ce problème pour tous les idéaux premiers de norme plus petite qu'une certaine borne B. Alors, en utilisant des résolutions de système linéaires nous pouvons résoudre le problème de l'idéal principal pour tout idéal principal qui est B-friable, c'est-à-dire tel que ses facteurs premiers sont plus petits que B. Ainsi, pour résoudre le problème en toute généralité, nous devons pouvoir réduire sa résolution pour un idéal $\mathfrak{a}$ de norme arbitraire à l'étude d'une ou plusieurs instances pour des idéaux B-friables[16].

3.4.3. *Descente vers des idéaux friables.* Soit $\mathfrak{a}$ un idéal principal. Au vu de la remarque précédente, nous allons chercher à *réduire* le problème de l'idéal principal sur $\mathfrak{a}$ à celui sur des idéaux friables. Pour ce faire nous allons nous efforcer de construire un élément $a \in \mathfrak{a}$ tel que $\mathfrak{a} \cdot \langle a \rangle$ soit suffisamment friable (pour une notion de suffisamment que nous éluciderons précisément au Chapitre 6). Ainsi, nous factorisons cet idéal en un produit d'idéaux $\mathfrak{b}_1 \cdots \mathfrak{b}_i$ de normes plus petites que celle de l'idéal de départ. Sur *chacun* de ces idéaux, nous pouvons recommencer la recherche d'un petit élément pour le rendre plus friable et recommencer jusqu'à arriver à la borne B introduite en paragraphe 3.4.2. La Figure 4 donne une vision schématique de cette phase que nous nommerons « descente ».

Une fois cette descente terminée, il suffit alors de résoudre le problème de l'idéal principal sur chacun d'entre eux, et de faire remonter les générateurs le long de l'arbre de descente pour en déduire un générateur de l'idéal de départ.

3.4.4. *Réduction d'un idéal* Afin de compléter cette présentation rapide de l'algorithme, il nous reste à éclaircir un point : étant donné un idéal $\mathfrak{a}$, comment construire un élément a tel que $\mathfrak{a} \cdot \langle a \rangle$ soit plus friable que l'idéal $\mathfrak{a}$? Pour ce faire, nous allons chercher à diminuer au maximum la norme de $\mathfrak{a} \cdot \langle a \rangle$. Cela revient à trouver un petit idéal principal $\langle a \rangle$ contenu dans $\mathfrak{a}$ et de diviser $\mathfrak{a}$ par $\langle a \rangle$. Ainsi, par multiplicativité de la norme, le quotient sera de norme plus petite que celle de $\mathfrak{a}$ et plus l'idéal $\langle a \rangle$ sera petit, plus le quotient le sera aussi. Il suffit donc de construire un élément a petit dans l'idéal, c'est-à-dire d'effectuer une *réduction du réseau algébrique* définie par $\mathfrak{a}$.

Nous sommes donc revenus à notre fil rouge : le problème d'arithmétique algorithmique se réduit *in fine* à une myriade d'instances de recherches de vecteurs courts de réseaux, tout comme la preuve du théorème des deux carrés se réduisait à la recherche d'un vecteur court dans un réseau bien choisi.

16 C'est à dire des idéaux n'ayant que des premiers de normes plus petites que B dans leurs décompositions en facteurs premiers

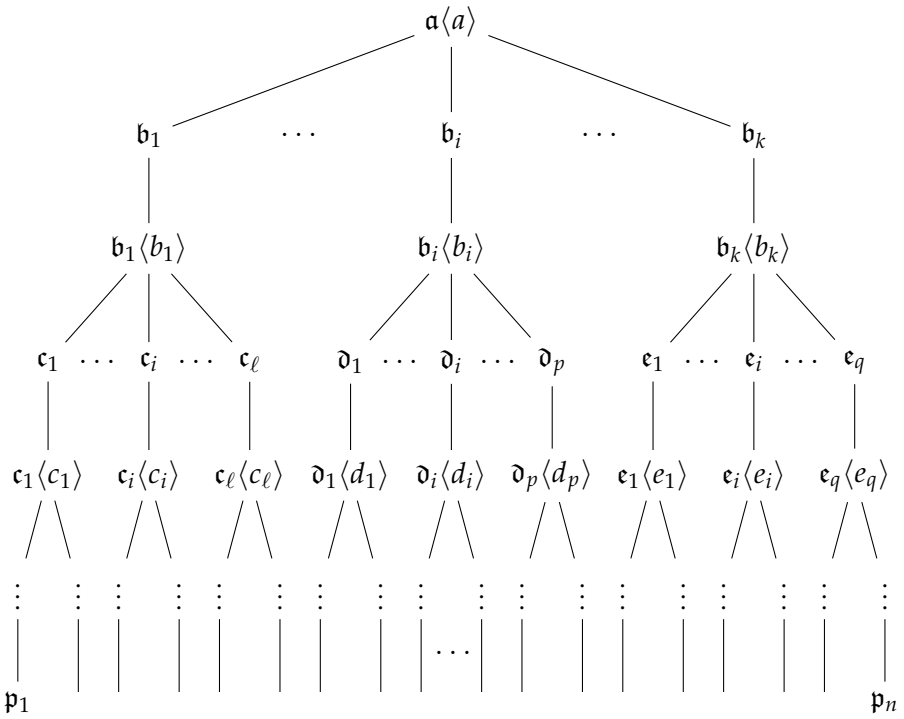


FIGURE 4 : Description schématique de la descente vers des idéaux de petites normes. La friabilité des idéaux est décroissante le long de l'arbre de descente. Les feuilles sont quant à elles toutes B-friables.

3.5 Ouverture cryptographique

Le XX^{e} siècle fut le théâtre de profonds changements en cryptologie. La cryptologie, étymologiquement la science du secret, ne peut en effet être vraiment considérée comme une science que depuis la fin des années 60, en lieu et place d'une simple discipline technique. Cette science englobe la cryptographie — l'art de protéger un secret – et la cryptanalyse – l'analyse, que l'on peut qualifier d'offensive, de cette dernière.

3.5.1. *Mutation du paysage cryptographique au cours du XX^{e} siècle.* Des prémices de la cryptographie, avec la fameuse *scytale* spartiate, jusqu'aux codes de guerre de la Première Guerre Mondiale, comme celui du télégramme de Zimmerman, la cryptographie était uniquement *manuelle* : tout chiffrement, tout déchiffrement et toute cryptanalyse reposait sur des transpositions de lettres, phonèmes ou mots réalisées à la main.

La cryptographie connaît un tournant juste après la fin de la guerre, avec la mise au point de la machine *Enigma* en 1919, lorsqu'un ingénieur hollandais, Hugo Alexander Koch, dépose un brevet de machine à chiffrer *électromécanique*. Il s'agit toujours d'un chiffre dit de transposition — chaque lettre est remplacée par une autre — avec la subtilité que la substitution change d'une lettre à l'autre. Le chiffrage est assuré par la machine, fonctionnant sur un astucieux système de rotors mobiles et de fiches électriques. La cryptanalyse s'est aussi mécanisée avec l'invention par Turing des *bombes*,

qui permettaient l'attaque par *force brute* d'Enigma, c'est-à-dire en essayant toutes les clefs possibles.

L'année 1976 marque profondément le fonctionnement des systèmes cryptographiques : c'est en effet de cette année-là que date la publication de l'article New Directions in Cryptography [44] de Diffie et Hellman. Cet article a introduit une méthode radicalement nouvelle pour distribuer les clefs cryptographiques, ce qui a résolu un des problèmes fondamentaux de la cryptographie : la distribution des clefs. Ce mode de distribution des clefs est appelé l'échange de clefs Diffie-Hellman. L'article a également stimulé le développement presque immédiat d'une nouvelle classe d'algorithmes de chiffrement, les algorithmes de chiffrement *asymétrique*. Avant cette date, tous les algorithmes de chiffrement (anciens et modernes) avaient été des algorithmes de chiffrement symétrique dans lesquels la même clef cryptographique était utilisée avec l'algorithme sous-jacent à la fois par l'expéditeur et le destinataire.

Quittons cet aparté historique pour renouer avec notre sujet d'étude, les réseaux.

3.6 La cryptographie à base de réseaux

La cryptographie asymétrique nécessite de trouver des problèmes difficiles, au sens de la théorie de la complexité. Mais au-delà même du problème il faut être en mesure de générer des instances difficiles[17] dudit problème pour pouvoir implémenter le cryptosystème.

Les premières méthodes utilisées en cryptographie pour produire des problèmes difficiles en cas moyen ont été proposées simultanément par Ajtai d'une part et par Hoffstein, Pipher et Silverman (NTRUSign) [81] d'autre part, en 1996. Toutes deux se basent sur les *réseaux Euclidiens*. Par la suite, Regev introduit en 2005 le problème de *l'apprentissage avec erreur* (LWE en anglais, pour *learning with errors*), accompagné d'une réduction quantique á un problème classique de réseaux, la recherche des vecteurs indépendants les plus courts. Dès lors, la cryptographie basée sur les réseaux a connu un vif essor et s'est imposée comme un des concurrents les plus sérieux pour l'avènement d'une cryptographie dite «post-quantique».

Il s'avère en effet que l'on ne parvient pas, à l'heure actuelle, à obtenir d'algorithme *quantique* efficace pour résoudre les problèmes difficiles sur les réseaux Euclidiens, alors que plusieurs des piliers de la cryptographie asymétrique, notamment le problème de la factorisation d'entiers et du logarithme discret dans les corps finis, s'effondreraient si nous disposions effectivement d'un calculateur quantique de grande envergure.

17 Un exemple bien connu est le problème 3-SAT de la satisfiabilité : les instances non construites spécifiquement pour être difficiles sont bien plus faciles à résoudre que ce que la NP-complétude du problème laisserait penser, ouvrant la porte à de nombreux algorithmes de résolution très efficaces en pratique.

Outre cet avantage de sécurité que confèrent les réseaux, ils ont aussi permis le développement de primitives cryptographiques riches, telles que le chiffrement homomorphe [62].

Ainsi, les réseaux Euclidiens constituent un socle particulièrement attractif pour le développement cryptographique, tant par les possibilités ouvertes que par la sécurité offerte.

3.7 Cryptanalyse de primitives par réduction de réseau

3.7.1. *Sécurité et réduction.* Afin d'évaluer la sécurité des primitives cryptographiques fondées sur les réseaux, il est souvent nécessaire d'utiliser des algorithmes de réduction de réseaux. En effet, la découverte d'un élément court dans un réseau utilisé pour construire un schéma cryptographique permet généralement de reconstruire la clef secrète.

Afin d'accélérer les cryptosystèmes sur les réseaux, il a été proposé, par exemple dans [82], [114], [105], de ne pas utiliser des réseaux arbitraires, mais plutôt des idéaux et plus généralement des réseaux sur des anneaux d'entiers de corps de nombres. En effet la structure algébrique de ces réseaux permet de calculer plus rapidement tout en minimisant la taille des objets cryptographiques manipulés[18].

Ainsi la sécurité de tels cryptosystèmes dépend de notre capacité à effectuer des réductions de réseaux sur des anneaux d'entiers, justifiant par là l'intérêt des techniques développées plus haut.

3.7.2. *Exemples pratiques utilisant la réduction de réseaux algébriques.* Le Chapitre 8, présentera les cryptanalyses de trois schémas basés sur les réseaux. La première est une application directe de la réduction rapide introduite au Chapitre 5 sur les fonctions multilinéaires de [58]. Partant de la clef publique, une base d'un réseau algébrique de rang 2 sur un corps cyclotomique, on retrouve un vecteur suffisamment court pour retrouver la clef secrète du schéma.

Ensuite, nous proposerons une attaque sur le chiffrement homomorphe de Smart et Vercauteren [153]. Sa clef secrète est constituée d'un petit élément d'un corps cyclotomique, et la clef publique correspondante est une $\mathbf{Z}$-base de l'idéal principal engendré par cette clef secrète. Ainsi, on peut attaquer à l'aide de l'algorithme de résolution du problème de l'idéal principal, comme présenté dans le Chapitre 6. Il est à noter que retrouver le générateur n'est pas suffisant, nous souhaitons en effet trouver un générateur *court*. Pour se faire nous utiliserons la réduction de [42] qui permet de transformer un générateur arbitraire en un plus court, et ce en temps polynomial.

Enfin nous détaillerons une attaque par canal auxiliaire sur la fonction de signature BLISS [49]. Une attaque par canal auxiliaire est une attaque qui recherche et exploite des failles dans l'implémentation, logicielle ou matérielle, d'un schéma cryptographique ou d'un protocole. Une telle attaque ne remet

18 Par exemple, un réseau provenant d'un idéal peut être représenté de manière compacte par deux polynômes, au lieu d'une matrice.

aucunement en cause la robustesse théorique d'un cryptosystème, mais expose seulement une faille dans son *implémentation*. Une analyse des consommations énergétiques du système embarquant la fonction de signature BLISS permet en effet d'estimer la norme de la clef secrète, qui est un petit élément d'un corps cyclotomique, sur un sous-corps. En utilisant la réduction de réseaux et les propriétés de factorisations des idéaux nous retrouvons la clef secrète à partir de sa norme sur un sous-corps, c'est-à-dire en résolvant une *équation de norme*.

PREAMBLE

1 GEOMETRY OF NUMBERS AND LATTICES

In his monograph *Geometrie der Zahlen* ([122]) — *Geometry of numbers* in English —, Hermann Minkowski had the fertile intuition that giving a geometric insight to number theory would allow to prove abstract results in a quasi-visual manner. This *geometry of numbers* permitted in particular to enhance the comprehension and to simplify results on units of number fields, as well as to extend the field of Diophantine approximation. A simple yet striking example of this new manner to "see" number theory is the famous *Fermat's two-squared theorem*, which can be presented as follows:

Theorem 1.1. *An odd prime number p can be written as the sum of two squares $x^2 + y^2$, with $x, y \in \mathbf{Z}$ if and only if $p \equiv 1 \mod (4)$.*

We give a quasi-intuitive proof after having introduced more formally the definition of lattice, which contrasts with the more classical proof using solely modular arithmetic reasoning. Since the early work of Minkowski, links between geometry and number theory have grown wider and deeper, spreading the fan of methods available to attack problems that look *a priori* purely arithmetics.

The central object of this geometrical theory of numbers is the notion of *lattice*, which formalizes the intuitive idea of *grid* in the plane or space. Through this manuscript, we are going to look at a generalization of the notion of lattice, which is no more living in the canonical Euclidean space but in spaces constructs from number fields, an algebraic generalization of the field of rational numbers. This tight bond between number theory, by essence arithmetic and algebraic, and geometry will be the guiding principle of this manuscript, viewed through threw prism of algorithmic reduction methods, allowing a computational approach of the theory developed by Minkowski.

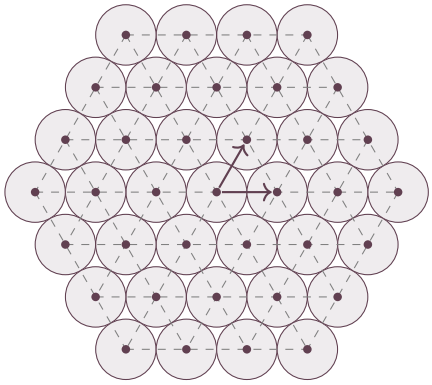
In order to forge an intuition on the lattice as a geometric object, we temporarily leave the arithmetic considerations to come back to the premises of this notion, finding their roots in a very visual and concrete problem.

1.1 Genesis: dense sphere packings

This is the problem of "sphere packing": what is the maximal density of a packing of hard spheres in $\mathbf{R}^n$ for a fixed dimension n? By sphere packing

of $\mathbf{R}^n$ we understand an infinite family of balls of same radius r and disjoint interiors, and by density, the fraction of the space $\mathbf{R}^n$ covered by these balls.

For instance, in dimension $n = 2$, the maximal density is reached by the so-called *hexagonal packing*, where the balls are centered on an hexagonal grid, as depicted below.

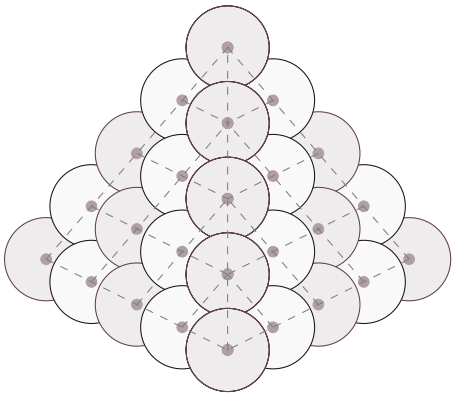


We can show that the density of this packing is

$$\frac{\pi}{2\sqrt{3}} \cong 0.9069,$$

by comparing the area of the surface defined by four circles in contact and by the area of the rhombus defined by the center of these circles. Even though the optimality of this packing is conjectured since the Pythagorean school, the first rigorous proof was only provided by Lagrange in 1773.

The dimension 3 case appears to be even more trickier. Kepler conjectured in 1611 that the traditional way of packing of fruits merchants and gunners. It consists of stacking the cannonballs by letting them slide onto the space formed by the ones of the floor below. This gives a kind of pyramid, represented below.



But it took almost four centuries to obtain the first correct proof of this conjecture of Kepler. It has been announced first by Hales in 1998, which have been fully published in 2005 and 2006 in [72, 73] The original rattle

of Hales is almost 300 pages long and required a large amount of computer-assisted computations. The certification and verification of these computations use, in particular, a technique called Interval arithmetic to detects and handle the rounding errors of numerical computations. We will stumble upon this technique, in our work this time, to ensure the certification of lattice reduction algorithms.

In both cases, it is interesting to remark that the center of the balls are placed in a *regular* manner in the space: they form a so-called "lattice".

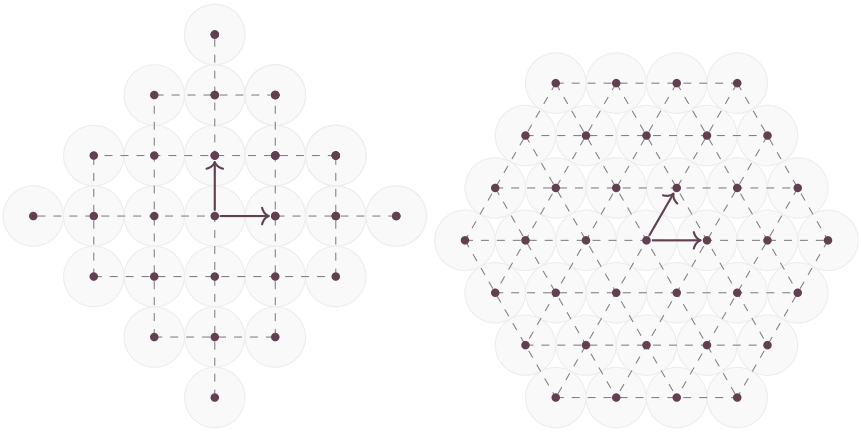
For higher dimensions, the question was open until the work of Viazovska in 2016, and of Cohn, Kumar, Miller, Radchenko, and Viazovska in 2017, who proved that the densest sphere packings in dimension 8 and 24 also come from the stacking of spheres on a lattice: the E_8 lattice in dimension 8 and the Leech lattice in dimension 24. It is noticeable that the proofs of these optimalities are way simpler and concise than the proof of Hales, but above all are not purely geometrical: the crux of their proof is the study of well-chosen modular forms. Dually to the ideas of Minkowski, this time, number theory is providing the tools to prove theorems in the realm of Euclidean geometry.

These very peculiar cases in dimensions 2,3,8 and 24 encourage the systematic study of the properties of packings of spheres centered on the points of lattices, simply named "lattice packings". It is one of the historical incentives of the study of Euclidean lattices. Hence, estimating the density of the densest lattice packing corresponds to estimate the exact value of the so-called *Hermite's constant* of the lattice, which we formally introduce in Paragraph 1.2.3. If the value of this constant in large dimension is still unknown[19], the lattice packings in small dimension have been fully determined: by Lagrange in dimension two [102], Gauss in dimension three [60], Korkine and Zolotareff in dimensions four and five [100], Blichfeldti for dimensions six to eight[18], and eventually Cohn and Kumar in dimension twenty-four [36]. The set of techniques introduced to prove these first results are prototypes of algorithms which are now known as *lattice reduction algorithms*.

1.2 Euclidean lattices

1.2.1. *Formalization of the lattice notion.* Through the examples given by sphere packings, we have seen that a lattice is a geometric structure of the Euclidean space $\mathbf{R}^n$, giving a regular grid. By the generic term *grid*, we understand that the vectors of the lattice regularly *pave* the space. To forge a visual intuition let us go back to the lattice of centers given by the hexagonal packing and the square packing in the plane $\mathbf{R}^2$:

19 For all known values, the densest sphere packings are indeed lattice packings, but there is no proof that this is the case in arbitrary dimension.



Since such a grid can be translated without changing its geometric properties we can suppose without loss of generality that the origin (0) belongs to the lattice. Thereof the sum of two lattice vectors is also a lattice vector, as well as the opposite of any lattice vector. Hence, a lattice is naturally endowed with a structure of *abelian group*. Besides, as the packings examples hinted, two vectors of a lattice can not be arbitrarily close to one-another: topologically speaking the set of vectors of a lattice is discrete for the Euclidean distance. It appears that these two notions actually encompasses exactly the properties of a lattice, defined in the following manner:

Definition 1.1. *A Euclidean lattice Λ is a discrete subgroup of $(\mathbf{R}^n,+)$, or equivalently it is the set of linear combinations with integral coefficients of a linearly independent family of vectors.*

Generally speaking, we denote by $\Lambda[v_1,\dots,v_k]$ the sub-lattice spanned by a linearly independent family $(v_1,\dots,v_k)$ of vectors, that is:

$$\Lambda[v_1,\dots,v_k] = \{a_1v_1+\cdots+a_kv_k \mid a_1,\dots,a_k \in \mathbf{Z}\}.$$

Therefore, a lattice is a *regular* structure—as the geometric arrangement of the neighbors of a vector is independent of the vector—and *discrete* of an Euclidean space.

1.2.2. *Bases and covolume.* A basis of a lattice Λ is a family $(v_1,\dots,v_k)$ of linearly independent vectors that spanned the whole lattice, that is such that

$$\Lambda[v_1,\dots,v_k] = \Lambda.$$

Clearly, a one-dimensional lattice is of the shape $\ell\mathbf{Z} = \{\ell k \mid k \in \mathbf{Z}\}$ for a certain real number ℓ, and thus has only two distinct bases ℓ and $-\ell$. However, as soon as the dimension of the lattice is greater than one, it possesses an infinite number of bases. They all share the same cardinality, which is called the "rank of the lattice". Bases are related to one-another by linear mapping which are *unimodular*: if $B = (b_1,\dots,b_n)$ is a basis of Λ, then a family $C = (c_1,\dots,c_n)$ of $\mathbf{R}^n$ is also a basis of Λ if and only if the $n \times n$ matrix U giving the coefficients of C in the basis B is invertible and has integral coefficients—since C is a basis of $\mathbf{R}^n$ constituted from lattice vectors—.

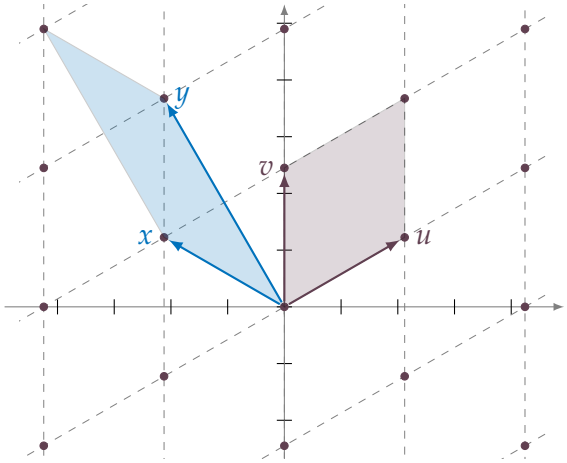


Figure 5: Plane lattice, with two basis $B = (u, v)$ and $C = (x, y)$ as well as their associated parallelograms $\mathcal{P}(B)$ and $\mathcal{P}(C)$. The area of these two parallelograms are equals.

It appears that these two conditions on U corresponds to the fact of being integral and having determinant ± 1.

In particular this implies that the determinant $\Delta(b_1, \dots, b_n)$ of the Gram matrix

$$\mathcal{G} = \left(\langle b_i, b_j \rangle\right)_{1 \leq i,j \leq n}$$

is a strictly positive real, independent of the basis B: this is the "discriminant" of the lattice. The "determinant" or "covolume" of a lattice Λ, denoted $\operatorname{covol} \Lambda$ is defined as the square root of the discriminant. It is equal to the volume — for the Lebesgue measure of $\mathbf{R}^n$ — of the parallelepiped defined by B in $\mathbf{R}^n$:

$$\mathcal{P}(B) = \left\{ \sum_{i=1}^{d} x_i v_i \,|\, 0 \leq x_i < 1 \right\}.$$

Figure 5 gives an instance of a plane lattice with two distinguished bases and their respective associated parallelepiped.

1.2.3. *Hermite's constant.* As dictated by the visual intuition, the periodicity and discreteness of a lattice ensure that these elements are not arbitrarily close to one another and in particular from the origin 0. Hence, it is possible to look for the shortest vectors of a lattice, which are the vectors which are the closest to 0. We call "first minimum" of a lattice Λ the quantity $\lambda_1(\Lambda) = \min_{x \in \Lambda \setminus \{0\}} \|x\|$.

The foundational work of Minkowski in his geometry of numbers ensures that a lattice vector always exists in any sufficiently large convex sets of the space:

Theorem 1.2 (Minkowsi)**.** *Let Λ be a lattice of $\mathbf{R}^n$ and $C \subseteq \mathbf{R}^n$ a measurable convex symmetric from o, such that*

$$\mathit{Vol}(C) > 2^n \operatorname{covol} \Lambda,$$

then C contains at least one non-zero point of Λ.

The intuition behind this theorem is relatively simple: we can convince ourselves by juxtaposing the parallelepiped formed by the 2^n possible orientations of the vectors of a basis B. The interior of this set is of volume $2^n \operatorname{vol} \mathcal{P}(B) = 2^n \operatorname{covol} \Lambda$, and by construction, it can not contain a lattice vector different from zero. Figure 2 gives an example of this gluing in dimension 2.

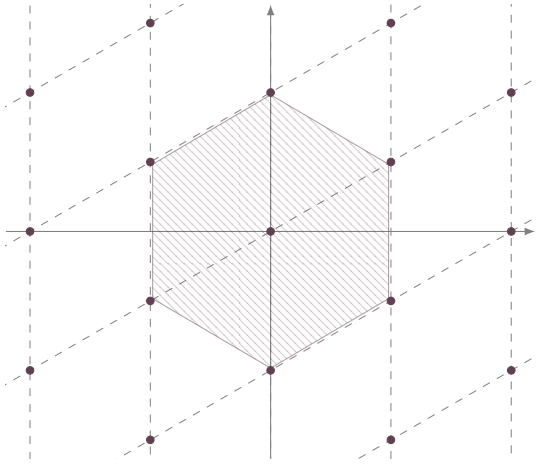


Figure 6: Example of the critical case for Minkowski's theorem.

If we apply this theorem with C being closed balls, we can prove that any n-dimensional lattice contains a non-zero vector whose length satisfies:

$$\|v\| = 2\left(\frac{\operatorname{covol}(\Lambda)\Gamma\left(\frac{n}{2}+1\right)}{\pi^{\frac{n}{2}}}\right)^{\frac{1}{n}} \leq \frac{1}{2}\sqrt{4+n}(\operatorname{covol}\Lambda)^{\frac{1}{n}},$$

with Γ the Euler Gamma function[20]. The right-hand side of this equation only depends on the dimension and of the covolume of Λ. Therefore, we can wonder what is the best upper-bound γ_n, such that for all lattice Λ of rank n there exists non-zero lattice vector such that we have $\|v\| \leq \gamma_n(\operatorname{covol}\Lambda)^{\frac{1}{n}}$. This corresponds to evaluate the constant:

$$\sqrt{\gamma_n} = \max_{\Lambda}\left[\frac{\lambda_1(\Lambda)}{(\operatorname{covol}\Lambda)^{\frac{1}{n}}}\right],$$

where the minimum is taken on the set of real lattices of rank n. This quantity is called the "n-dimensional Hermite's constant". Since each lattice, Λ induces a sphere packing of $\lambda_1(\Lambda)/2$ with centers the lattice vectors, it appears that estimating the numerical value of Hermite's constant allows to estimate the density of the densest lattice packing. Hence, the exact computation of this constant is one of the principal problems of the geometry of numbers.

It is interesting to remark that the results presented ensure that a short vector *theoretically* exists. However, in practice, its construction remains

20 The volume of the n dimensional ball of radius r is indeed equal to $\frac{\pi^{n/2}r^n}{\Gamma\left(\frac{n}{2}+1\right)}$.

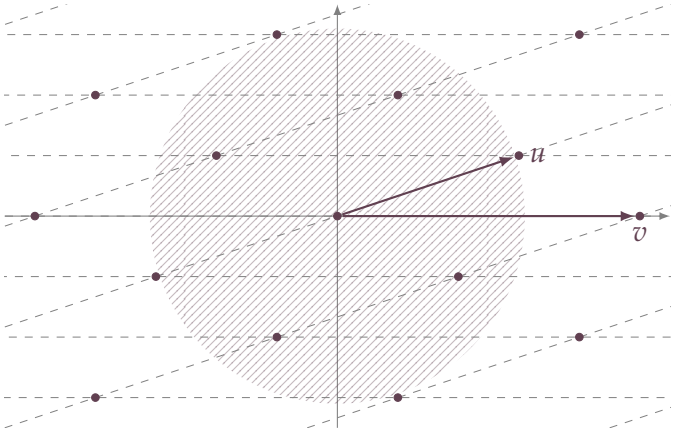


Figure 7: Example of the lattice obtained for $p = 5$, with the disk C hatched.

hard: the algorithmic search of the shortest vector of an arbitrary lattice is a hard problem in the sense of complexity theory[21]. As such, for *practical* purposes, when the dimension is too large, we do not look for the shortest vector but instead for an *approximation*, that is to say for a non-zero vector located in a ball a sufficiently small radius. The algorithmic methods to solve this problem are the basis of the so-called "lattice reduction".

1.2.4. *Digression: back on the two-square theorem.* Before pursuing our journey in the world of lattices and their reduction, let us go back for a moment on our introductory theorem. Indeed, we can now provide a concise proof of Fermat's two square theorem by using the notion of lattices, in particular using Minkowski's theorem.

Let then p be a prime congruent to -1 modulo 4, so that -1 is a square modulo p: hence it exists an integer q such that $-1 \equiv q^2 \mod p$. Let us consider the plane lattice Λ spanned by the vectors $u = (q, 1)$ and $v = (p, 0)$. Denote by M the matrix $[u, v]$. The covolume of this lattice is by definition :

$$\operatorname{covol} \Lambda = \sqrt{\det(M^T M)} = p$$

Let C be the convex symmetrical body defined by

$$C = \left\{(x, y) \in \mathbf{R}^2 \,\middle|\, x^2 + y^2 < 2p\right\},$$

that is the open disk or radius $\sqrt{2p}$, centered in 0. Its volume is then: $\operatorname{vol} C = 2p\pi > 4p = 2^2 p = 2^2 \operatorname{covol} \Lambda$.

Hence, by Minkowski's theorem, there exists a non-zero lattice vector in this ball, denoted by (x, y). But then, there exists $a, b \in \mathbf{Z}$ such that : $(x, y) = au + bv = (aq + bp, a)$. Thus, by taking the square of the norm of this vector we get:

$$x^2 + y^2 = a^2 + a^2q^2 + b^2p^2 + 2abpq \equiv a^2(1 + q^2) \mod p.$$

21 The search problem of the shortest vector is indeed a NP-hard problem, as shown by Peter van Emde Boas in [51].

But since $q^2 \equiv -1 \mod p$ we have $x^2 + y^2 \equiv 0 \mod p$, and so $x^2 + y^2 = p$ as $x^2 + y^2 \neq 0$ and $x^2 + y^2 < 2p$. Figure 7 gives an example of the setting for the case $p = 5$. We can see the lattice vectors inside the disk C, allowing to retrieve the square factors as in the proof.

This duality between arithmetic and geometry is our leitmotiv all along this manuscript. We wished to develop this topic through the prism of the algorithmic reduction theory, a more recent matter which finds its roots in the work of Gauss and Lagrange in the XIX$^{\text{th}}$ century.

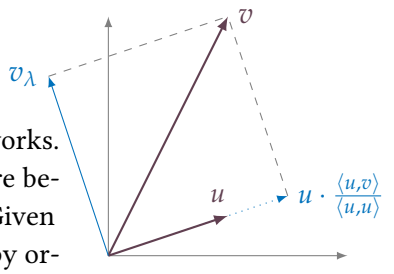
1.3 On the algorithmic reduction of Euclidean lattices

We have seen that lattices possess an infinite number of bases as soon as the dimension is greater than one. We also know that an Euclidean vector space has privileged bases: the orthogonal bases. Indeed, the (local-) compactness of $\mathbf{R}^n$ allows to straighten iteratively each vector of a basis to ensure that it becomes orthogonal to the one before it.

1.3.1. *Orthogonalization.* To be more precise, let us look at how to orthogonalize a vector v with regard to a vector u, by using only a linear combination of u. Let λ be a real parameter and $v_\lambda = v - \lambda u$ the vector which we want to make orthogonal to u.

From the equation $\langle u, v_\lambda \rangle = 0$ we get:

$$\langle u, v \rangle - \lambda \langle u, u \rangle = \langle u, v_\lambda \rangle = 0,$$



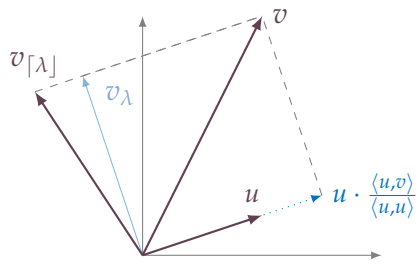
so that choosing $\lambda = \langle u, v \rangle / \langle u, u \rangle$ works. This construction is illustrated in the figure beside. This situation can be generalized. Given a family $(v_1, \ldots, v_n)$ of vectors, we start by orthogonalizing v_2 with v_1. Denote by v_2^* this new vector. Then we can orthogonalize v_3 with regards to v_1 and v_2*, yielding a vector v_3^*, which is orthogonal to these two vectors. We can go on with v_4 by making it orthogonal to v_1, v_2^*, v_3^*, and so on. This algorithm allows constructing an orthogonal basis of the space so that the subspace spanned by its i-th first factors is the same as the subspace spanned by $v_1, \ldots, v_i$. This method is the "Gram-Schmidt orthogonalization process". Its pseudo-code translation is given in Algorithm 4:

Algorithm 4 — Gram-Schmidt

Input : $(v_1, \dots, v_n)$ an independent family of vectors
Output : An orthogonal family $(v_1^*, \dots, v_n^*)$.

$v_1^* \leftarrow v_1$
for $k = 2$ **to** n **do**
  $v_k^* \leftarrow v_k$
  for $j = 1$ **to** $k-1$ **do**
    $v_k^* \leftarrow v_k^* - \left(\frac{\langle v_k, v_j^* \rangle}{\|v_j^*\|^2} \right) \cdot v_j^*$
  end for
end for
return $(v_1^*, \dots, v_n^*)$

1.3.2. *Size-reduction.* However, in the lattice case, the rigidity imposed by the discreteness generates an obstruction to this construction: since the only linear combinations allowed by the structure are the combinations with integral coefficients, we can only *approximate* the vectors given by the orthogonalization process.



Indeed, if we go back to the case presented earlier with the vectors U and v. then we can not search for an *arbitrary* linear combination of them, but only for integral linear combinations. Thus it is not possible to make u_λ orthogonal to u. Instead, we can still try to minimize the orthogonality defect, that is finding the integral values of λ minimizing the function $\lambda \mapsto \langle v_\lambda, u \rangle$, that is the integer the closest to $\lambda = \langle u, v \rangle / \langle u, u \rangle$. Similarly to the Gram-Schmidt process, we can reduce iteratively a family $(v_1, \dots, v_n)$ of lattice vectors, by *discretizing* the operations made at each step. Instead of making a vector orthogonal to the previous ones, we aim at reducing as much as possible the orthogonality defect. It appears that this minimization also shrinks the size of the considered vectors. Hence this process is called "size-reduction" or "weak reduction". It is described in pseudo-code in Algorithm 5.

Algorithm 5 — Size-reduction

Input : $(v_1, \dots, v_n)$ a family of vectors
Output : A size-reduced family $(v_1, \dots, v_n)$.

$v_1^*, \dots, v_n^* \leftarrow$ **Gram-Schmidt**$(v_1, \dots, v_n)$
for $k = 2$ **to** n **do**
  for $j = i - 1$ **to** 1 **do**
    $v_k \leftarrow v_k - \left\lceil \frac{\langle v_k, v_j^* \rangle}{\|v_j^*\|^2} \right\rfloor \cdot v_j$
  end for
end for
return $(v_1, \dots, v_n)$

1.3.3. *Densify the successive sublattices.* Hence, the notion of privileged bases seems less canonical in the lattice context than in a Euclidean space. As we have seen, the size-reduction allows reducing the orthogonality defect and the norm of vectors of a basis. We can thus try to construct a basis with vectors as short as possible. To do so we can, for instance, strive to *densify* the successive sublattices spanned by the basis vectors, i.e. to shrink their volumes. More specifically, let us fix a basis $(v_1, \dots, v_n)$ of a lattice Λ, and denote by Λ_i the sublattice spanned by the i-th first vector's $v_1, \dots, v_i$. Suppose that Λ_1 is the densest sublattice among all the sublattices of rank one. Therefore, v_1 is necessarily the shortest vector of Λ. Then, if Λ_2 is also the densest sublattice among the sublattices of rank 2 containing v_1, then v_2 is also presumably short, without what the covolume of this sublattice would be large. The same goes for the following sublattices Λ_i: having dense sublattices Λ_i implies getting somewhat short vectors v_i.

1.3.4. *Towards LLL reduction.* Patently, the size-reduction acts on vectors without modifying the sublattices Λ_i. If we aim at shrinking the covolume of these sublattices, we thus need to *permute* the basis vectors. Of course, it would be very costly to enumerate all the $n!$ permutations of these n vectors. However, remember that the symmetric group is generated by the transpositions of the shape $(i, i+1)$, that is, the transpositions of two successive elements. As such, we can try to densify the covolumes of the Λ_i by successively swapping two consecutive vectors of the basis. Once a pair of them have been swapped, it is now possible to perform a new pass of size-reduction. This procedure can be pursued up-to no new change, either being swaps or size-reduction, improve the basis. Then, such a basis is said to be *reduced*. This simple sketch translates easily in pseudo-code:

1. **While** a modification improves the basis **do**
2. Size-reduce the basis

3. **If** it exists an index $1 \leq i \leq n$ such that the covolume of the sublattice spanned by $v_1, \ldots, v_{i-1}, v_{i+1}$ is smaller than the covolume of the sublattice spanned by $v_1, \ldots, v_{i-1}, v_i$ **then** exchange the vectors v_i and v_{i+1}.

4. **end**

By keeping track of the position of the last exchange, we can avoid restarting from the beginning at each iteration. Besides, we relax the exchange condition by a parameter $0 < \delta < 1$. This modification enforces a *polynomial* running time of the algorithm. All in all, we obtain Algorithm 6. This algorithm is exactly[22] the Lenstra-Lenstra-Lovàsz (LLL) algorithm, introduced in 1982 with application the factorization of polynomials with integral coefficients. A basis is called LLL-reduced is it satisfies the two following properties:

1. It is size-reduced.

2. No swap can reduce the covolume of one of the successive sublattices $\Lambda[v_1], \Lambda[v_1, v_2], \ldots, \Lambda$.

Hence, the LLL algorithm returns a LLL-reduced basis from an arbitrary basis of a lattice. If we denote by B an upper-bound on the number of bits required to store any coefficient of a basis of a lattice of rank n, then the LLL reduction runs in a $O(n^5 B^3)$ binary operations. Variants of this algorithm, such as [128] which uses floating-point arithmetic, allows reducing such a lattice in only a $O(n^4 B \log B)$ operations.

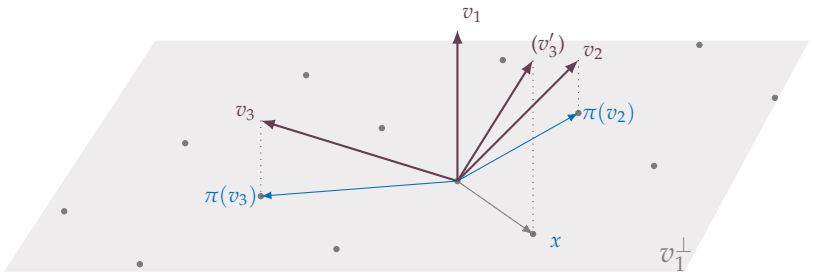
Algorithm 6 — LLL-reduction

Input : $(v_1, \ldots, v_n)$ a basis of a rank n lattice Λ
Output : A LLL-reduced basis of Λ

$k \leftarrow 2$
while $k \leq d$ **do**
  $v_1^*, \ldots, v_n^* \leftarrow$ **Gram-Schmidt**$(v_1, \ldots, v_n)$
  for $j = k - 1$ **to** 1 **do**
    $v_k \leftarrow v_k - \left\lceil \frac{\langle v_k, b_j^* \rangle}{\|b_j^*\|^2} \right\rfloor \cdot v_j$
  end for
  if $\mathrm{vol}(v_1, \ldots, v_{i-1}, v_{i+1}) < \delta \,\mathrm{vol}(v_1, \ldots, v_{i-1}, v_i)$ **then**
    $k \leftarrow k + 1$
  else
    Swap v_k et v_{k-1}
    $k \leftarrow \max(k - 1, 2)$
end while
return $(b_1, \ldots, b_n)$

22 The reader who is already familiar with lattice reduction shall remark that the exchange condition with covolumes is exactly the so-called Lovàsz condition on the norm of projected vectors.

The following representation illustrates a step of the LLL algorithm in dimension 3. The current basis is v_1, v_2 , v_3. We aim at reducing v_3 using the vector v_2. This step is going to be performed by *lifting* the result , denoted x, of the reduction of $\pi(v_3)$ by $\pi(v_2)$, where π is the orthogonal projection onto the subspace $v_1^\perp$. This lifting is done by choosing the element v_3' of $\Lambda \cap (x + v_1)$ of smallest norm.



1.3.5. *Approximation factor and reduction with oracles.* The LLL algorithm is a *greedy* algorithm as it swaps two vectors as soon as an improvement on the covolumes is possible. However, it is not optimal with regards to the length of the first vector of the basis nor the density of the successive sublattices. Indeed, it gives an *approximation* of the optimal values of these quantities, quantified by the following lemma:

Lemma 1.1. *Let $(v_1, \ldots, v_n)$ a basis of lattice Λ, which is reduced using the LLL algorithm with parameter $0 < \delta < 1$. Then for each of the $1 \leq k \leq n$, we have:*

$$\operatorname{covol} \Lambda[v_1, \ldots, v_k] \leq \left(\delta - \frac{1}{4}\right)^{-\frac{(n-k)k}{4}} \operatorname{covol} \Lambda^{\frac{k}{n}}$$

For instance, this lemma ensures that the LLL algorithm can be used to find a vector of norm smaller than: $\left(\delta - \frac{1}{4}\right)^{-\frac{n-1}{4}} \operatorname{covol} \Lambda^{\frac{1}{n}}$. Knowing that Hermite's constant γ_n is polynomial in the rank n, it is then an exponential approximation of the shortest vector. It is then natural to wonder if it is possible to shrink this exponential factor in $(\delta - 1/4)^{-k(n-k)/4}$. But is so, for which computational cost? Indeed these so-called "approximation factors" can be reduced, if we have access to an oracle which can determine exactly the shortest vector of an arbitrary lattice. The half-reduction algorithm, introduced by Schnorr [145] in 1987, can be seen as a generalization of the LLL algorithm, in which we do not look for a short vector in a projected rank 2 sublattice, but this time in a rank β projected sublattice, for a parameter $1 < \beta < n$. This algorithm yields vectors shorter than the vectors discovered by the LLL algorithm, but has a complexity exponential in β. Hence, there is a tradeoff between the quality of the reduction, essentially given by the length of the first vector of the basis and the time required to perform the computation. The most striking illustration of this tradeoff is given in the following theorem, coming the DBKZ variant of Micciancio and Walter [121], combined with the enumeration techniques of of[11] for the shortest vector oracle:

Theorem 1.3. *Their exists an algorithm which outputs a vector v satisfying*

$$\|v\| \leq \beta^{\frac{n-1}{2(\beta-1)}} \cdot (\operatorname{vol}\Lambda)^{\frac{1}{n}},$$

for a parameter $0 \leq \beta \leq n$ The complexity of the corresponding computation is a $\mathrm{Poly}(n, \log B)\left(\frac{3}{2}\right)^{\beta/2+o(\beta)}$, where B is a bound on the bit size of the entree.

2 THE ALGEBRAIC SETTING: LATTICE AND NUMBER FIELDS.

2.1 Number fields, algebraic numbers

The concept of algebraic number, introduced by Niels Abel, finds its origin in the willing of solving "algebraic equations", that is of solving equations of the shape $P(x) = 0$, where P is a rational polynomial. The fundamental idea, hinted in the letters of Évariste Galois, consists in adjoining a root α of P to the rationals $\mathbf{Q}$ and to study the resulting object:

$$\mathbf{Q}(\alpha) = \{q_0 + q_1\alpha + \cdots + q_k\alpha^k \mid k \leq 0, q_1, \ldots, q_k \in \mathbf{Q}\}.$$

Let us give an example of number fields which is going to be used to exemplify the notions we introduce subsequently: we look at the field $\mathbf{Q}(i)\{a + ib \mid a, b \in \mathbf{Q}\}$ coming form adjoining the roots i and $-i$ of the polynomial $X^2 + 1$ to the rational numbers.

2.1.1. *Structure of a number field.* It appears that $\mathbf{Q}(\alpha)$ is a field, and as such a field extension of the rationals. Moreover, it is of finite degree—its dimension $[\mathbf{Q}(\alpha) : \mathbf{Q}]$ as $\mathbf{Q}$-vector space is finite—over $\mathbf{Q}$. Henceforth, each element of $\mathbf{Q}(\alpha)$ is *algebraic*, that is annihilated by a rational polynomial. If we limit ourselves to looking at the elements of $\mathbf{Q}(\alpha)$ which are roots of *unitary* polynomials with *integral* coefficients, we get a subset $\mathscr{O}_{\mathbf{Q}(\alpha)}$, which is a ring for the laws of $\mathbf{Q}(\alpha)$. It is the *ring of integers* of the field $\mathbf{Q}(\alpha)$, which is a generalization of the relative integers in the sense that the ring of integers of $\mathbf{Q}$ is $\mathbf{Z}$. It also constitutes a n-dimensional generalization of $\mathbf{Z}$ as we can prove (see Theorem 1.2.2) that there exists elements $v_1, \ldots, v_n \in \mathscr{O}_{\mathbf{K}}^n$ so that $\mathscr{O}_{\mathbf{K}}$ is the set of linear combinations with coefficients in $\mathbf{Z}$ of the v_i, that is to say:

$$\mathscr{O}_{\mathbf{K}} \cong v_1\mathbf{Z} \oplus \cdots \oplus v_n\mathbf{Z} \cong \mathbf{Z}^n.$$

If we go back to our example the ring of integers of $\mathbf{Q}(i)$ is $\mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$ called the ring of *Gaussian integers*. It is clear that $\mathbf{Z}[i] \cong \mathbf{Z}^2$ as an abelian group.

2.1.2. *Euclidean norm in a number field.* A number field $\mathbf{K} = \mathbf{Q}[\alpha]$ can be *embedded*[23] in the field of complex numbers $\mathbf{C}$ in various manners. Since the field $\mathbf{K}$ is spanned by an element α, any embedding σ is exactly determined

23 In this context, an embedding is *algebraic*, that is which preserves the properties of $\mathbf{K}$: it is nothing else than a field morphism.

by the image of α. Remembering that any polynomial $P \in \mathbf{Q}[X]$ we have $\sigma(P(\alpha)) = P(\sigma(\alpha))$, then $\sigma(\alpha)$ is a *conjugate* of α, that is is also a root of the defining polynomial of $\mathbf{K}$. Hence there is at most $n = [\mathbf{K} : \mathbf{Q}]$ possible embeddings and we can check that they are all distinct. We denote them by $\sigma_1, \dots, \sigma_n$.

This embeddings allows to define the *Archimedean embedding* of the field $\mathbf{K}$ in $\mathbf{C}$:

$$\sigma : \left| \begin{array}{ccc} \mathbf{K} & \longrightarrow & \mathbf{C}^n \\ x & \longmapsto & (\sigma_1(x), \dots, \sigma_n(x)) \end{array} \right. .$$

Then, we can *lift* the canonical Hermitian structure of $\mathbf{C}^n$ to $\mathbf{K}$ by defining the inner product

$$\langle a, b \rangle = \sum_{i=1}^{n} \sigma_i(a)\overline{\sigma_i(b)},$$

where $z \mapsto \overline{z}$ is the usual conjugation of $\mathbf{C}$. This inner product embed $\mathbf{K}$ by its so-called *canonical norm*, given by $\|a\| = \sqrt{\langle a, a \rangle}$.

In our example, there exists two embeddings σ_1 and σ_2 respectively sending the root i to itself and on its conjugate $-i$. Thus the Archimedean embedding of an element $a + ib \in \mathbf{Q}$ is the vector $(a + ib, a - ib) \in \mathbf{C}^2$. The corresponding algebraic norm of $a + ib$ is then $(a + ib)(a - ib) = a^2 + b^2$.

But then, what is the connection with the lattices, introduced over?

2.2 Natural structure of lattice of the ring of integers

Since the ring of integers $\mathscr{O}_\mathbf{K}$ of $\mathbf{K}$ is included in $\mathbf{K}$, it inherits the Euclidean structure that we just defined. Hence, as $\mathscr{O}_\mathbf{K}$ is the set of integral linear combinations of a basis, it is a lattice in the sense given by Paragraph 1.2.1.

2.3 Algebraic lattices

We introduced lattices as a set of integral linear combinations living in a Euclidean space. The algebraic structure formally describing the set of such linear combinations is the structure of *free* $\mathbf{Z}$*-module*.

2.3.1. *A little digression on modules.* Let us recall briefly that a module is an algebraic structure defined similarly to a vector space, but in which the field of scalars is replaced by a ring. It is thus an additive group endowed with an external multiplication law with the elements of a ring, with the right compatibility laws. This structure is more subtle than the vector space and the classification of modules is a lot more complicated. Nonetheless, a *finitely generated free* module over a ring R behaves quite similarly to a vector space. In particular, the module possesses bases of equal cardinality. As such it is isomorphic to R^n for a certain n, in the same way, a n dimensional $\mathbf{K}$-vector space is isomorphic to $\mathbf{K}^n$. Hence a free $\mathbf{Z}$-module of rank n is nothing else than a "vector space" over $\mathbf{Z}$, which is a structure of the shape $b_1\mathbf{Z} \oplus \cdots \oplus b_n\mathbf{Z}$ for $b_1, \dots, b_n$ being a basis.

2.3.2. *Generalizing the notion of lattice.* Henceforth, we can reinterpret the definition of a lattice. It is the datum of two structures: on the one hand, the *algebraic structure*, given by the $\mathbf{Z}$-module, and on the other hand the *metric* structure, given by the Euclidean norm on its elements. We now aim at generalizing the notion of lattice by replacing the ring $\mathbf{Z}$ by more general lattices.

A natural family of candidates for these rings is the class of ring of integers of number fields. These rings have an arithmetic which is quite close to the usual arithmetic of $\mathbf{Z}$. Hence let us fix a number field $\mathbf{K}$ and denote by $\mathscr{O}_{\mathbf{K}}$ its ring of integers. From a metric point of view, we have access to the canonical norm of $\mathbf{K}$, which extends to $\mathbf{K}^n$ for any integer n. From now on we are not going to look at modules over $\mathbf{Z}$ but over $\mathscr{O}_{\mathbf{K}}$. Hence a $\mathscr{O}_{\mathbf{K}}$-lattice of rank n is an $\mathscr{O}_{\mathbf{K}}$-module[24] of rank n endowed with the Euclidean norm of $\mathbf{K}^n$.

The works presented in this manuscript aims at extending the lattice reduction algorithms to lattices over number fields.

3 CONTRIBUTIONS

3.1 The issue of provable and certifiable reduction

We have seen that we could construct a metric structure adapted to a lattice over a number field by using the embedding into the field of complex numbers. It is useful to remark that these embeddings are computed as polynomial evaluations over the roots of the defining polynomial of the number field. Hence, once these roots are approximated, for instance by refinements of Newton's iterative method [69], the computations of inner products between lattice vectors are also approximation of their actual value. If we launch the LLL reduction algorithm on this approximation of the lattice we can not enforce that the returned basis is indeed LLL-reduced.

3.1.1. *An interval arithmetic* The interval arithmetic is a computational method consisting of manipulating intervals to represents numbers (such as integers or floating-point numbers), to obtain rigorous results. This approach allows to bound the rounding errors in the approximated computations and thus to develop certifying numerical methods, yielding reliable results.

If the techniques to round real numbers belong to the common folklore of numerical computations—Archimedes was already such tools for his first estimates of the constant π in the III$^{\text{th}}$ century bf. J.-C.—, the formalization of under some precise arithmetic rules only comes back to the work of Young [165] in 1931. Subsequently, an article on the algebra of intervals for numerical analysis was published by Sunaga in 1958 [158]. A landmark in the birth of modern interval arithmetic is the book *Interval Arithmetic* of Ramon E. Moore in 1966 [125], widening its range of applications. From a very

24 For compatibility reasons which we elucidate in Chapter 3, we will see that it is too restrictive to only consider free modules, but that relaxing this condition in projectivity is sufficient.

simple principle, interval arithmetic gives a generic method to estimate the rounding errors and imprecision mistakes made at runtime.

The crux of the method it to represent a number $x \in \mathbf{R}$, by an interval $[x^-, x^+] \subset \mathbf{R}$, such that $x \in [x^-, x^+]$. Thus we can add, subtract, multiply, and divide intervals while still enforcing the certification that the result of the operation lies between the bounds of the resulting interval. The usefulness of this arithmetic is to certify comparisons between numbers and to detect during the computation if too many approximations happened to conclude. Let us look at a schematized example

The interval reprinting x ensure that $x < 0$ since the upper bound e x^+ is also smaller than o and that by definition, $x \leq x^+$. On the contrary, one can not conclude that e $y < z$, since $y^+ > z^-$, which implies that the intervals containing y and z are overlapping. But we can still infer at this point that the approximations leading to these representations of the values y and z are too large to certify the global result $y < z$ and as such that the computation must be run again with more precision.

3.1.2. *Lattice reduction in interval arithmetic.* The fast implementations of the LLL algorithm use floating-point arithmetic with low precision to speed-up the computations. Of course, bounds on the minimum amount of precision to use are provable, such for instance the L^2 variant of Nguyen and Stehlé [131]. These algorithms suppose that the lattices are given exactly, that is that the inner products are computed table and representable at arbitrary precision, and in particular at the precision used in the algorithm. However it is impossible to compute *a priori* the inner products at this precision, the bounds upper-evoked are not sufficient to reduce the lattice. We must be able to detect that the approximation used in the representation of the lattice is not sufficient.

These algorithms are designed with the promise that the lattices are given explicitly, that is that the inner products are computable and representable at the precision used by the reduction algorithm. However, it is impossible to know in advance the error made on these inner products, we can not use such bounds as we need to determine if the precision used to represent the lattice is actually sufficient. This obstruction is fully resolved by the use of interval arithmetic: if a lack of precision is detected at runtime by the algorithm and the precision of the internal quantity is sufficient, then we can conclude that the representation of the lattice—in the case of an algebraic lattice, of the embedding of lattice—is not sufficient and that it is needed to recompute it with greater accuracy. Hence this technique allows computing in a certified manner reduced bases of an algebraic lattice, by using only its metric structure: the reduction of an $\mathcal{O}_\mathbf{K}$-lattice of rank d over a number field of degree n is then done by the reduction of an (approximate) $\mathbf{Z}$-lattice of rank $d \times n$.

3.1.3. *A certified reduction with interval arithmetic.* An interesting byproduct of the reduction with interval arithmetic lies in the fact that it allows certifying the computation *at any moment of the reduction*, and in particular that the computation is the same as a LLL reduction using *exact* arithmetic. In particular this opens the door to an experimental study of the exact version of LLL, with the speed of floating-point implementation but with the promise that the execution trace is still the same as the exact version.

Lattice reduction in the average case, that is the study of reduction algorithms on random lattices[25], is not well understood. For instance, the practical results of [130] ensure that a LLL reduced basis is far more reduced than the theoretical bound for the worst case. However, the theoretical results of [95] prove that the majority of LLL-reduced bases are bases which reach the worst-case bounds. Hence the LLL algorithm naturally *choose* bases of better quality. Therefore it is very interesting to have access to a fast reduction algorithm having the same execution trace as original LLL to pursue extensive practical studies.

3.2 Towards a faster reduction of algebraic lattices

Thanks to interval arithmetic, it is possible to ensure a proved reduction for generic lattices and in particular for algebraic lattices. However, the reduction of $\mathcal{O}_{\mathbf{K}}$-lattices starts by *descending* the $\mathcal{O}_{\mathbf{K}}$-lattice over $\mathbf{Z}$. This corresponds to forget the algebraic structure of $\mathcal{O}_{\mathbf{K}}$ and running the LLL reduction. But the image over $\mathbf{Z}$ of a rank d $\mathcal{O}_{\mathbf{K}}$-lattice is of rank $d \times n$, where n is the degree of $\mathbf{K}$. Hence, even in the case where the lattice is of small rank over $\mathcal{O}_{\mathbf{K}}$, the reduction can be very costly as the dimension over $\mathbf{Z}$ might be large.

This process completely forgot about the algebraic specificities of the ring $\mathcal{O}_{\mathbf{K}}$. But these properties translate into symmetries over s $\mathcal{O}_{\mathbf{K}}$-modules. Consequently, the above-mentioned reduction can not take these symmetries into account. Thus, it is natural to wonder if it is possible to *exploit* the algebraic structure of $\mathcal{O}_{\mathbf{K}}$ to speed up the reduction.

3.2.1. *Reduction on Euclidean number fields.* A first step towards a generic algorithmic for algebraic lattices consists in studying the reduction on rings where the arithmetic is the closest to the arithmetic of $\mathbf{Z}$. A quick look at the classical taxonomy of rings reveals that the so-called "Euclidean" rings are the closest to the relative integers. For such rings the LLL-reduction can be straightforwardly adapted, with almost no modifications[26]. This variation has been introduced by Napias in [126], and the improved by Camus in [30].

Even though this reduction is simple and allows to effectively reduce algebraic lattices, we are still very far from the general case, as the norm Eu-

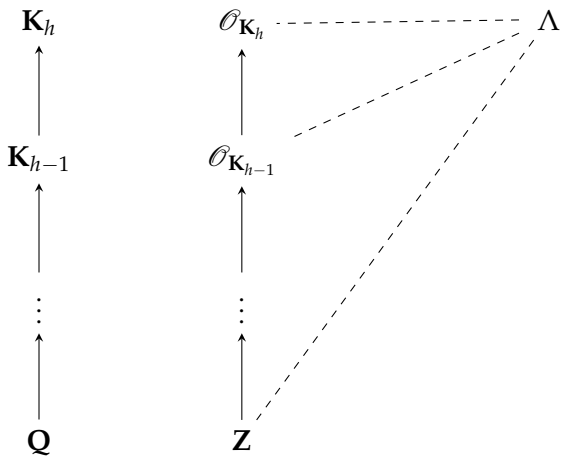
25 It is possible to define formally a notion of random lattice, by renormalizing the Haar measure on the moduli space of lattices

26 Indeed, it suffices to replace the absolute value over $\mathbf{Q}$ by the algebraic norm of the ring.

clidean rings are very rare[27]. Let us look at a more general case, which is studied in detail in Chapter 5.

3.2.2. *Recursion in a tower of number fields.* Let us consider a number field s $\mathbf{K}_h$, on the top of a *tower* of number fields: $\mathbf{Q} \subseteq \mathbf{K}_0 \subseteq \cdots \subseteq \mathbf{K}_{h-1} \subseteq \mathbf{K}_h$, as well as an $\mathcal{O}_{\mathbf{K}_h}$-lattice Λ.

Since we can see Λ as a lattice over any of the intermediate ring of integers $\mathcal{O}_{\mathbf{K}_i}$, the existence of such a tower allows to consider a naturally *recursive* approach of the reduction. Indeed, we know how to reduce any lattice over $\mathbf{Z}$, ring which is on the bottom of the tower and which constitutes *de facto* a base case for our algorithm.



3.2.3. *General structure of the reduction.* The general structure of our recursive reduction algorithm is close to the one of the classical LLL: in addition to the pass of size-reduction which allows handling the size of the coefficients during the computation, the crux of the algorithm is the reduction of an orthogonally projected lattice of rank 2. We keep here this blueprint, which enables the extension of the reduction for rank two lattices to arbitrary lattices.

The reduction hence boils down to reduce arbitrarily $\mathcal{O}_{\mathbf{K}_h}$-lattice, say Λ of rank 2, corresponding to the orthogonal projection of a pair of vectors of the current basis. To do so, we make use of the recursive structure of the tower of number fields to "see" the lattice Λ as an $\mathcal{O}_{\mathbf{K}_{h-1}}$-lattice of rank $2 \times [\mathbf{K}_h : \mathbf{K}_{h-1}]$ and to recursively call the reduction on this new lattice. At each recursive call, we go down in the tower, so that the leaves correspond to the reduction of a Euclidean ring[28], where we know how to perform the reduction and yield a reduced lattice.

27 Among the known Euclidean ring, let us mention the quadratic imaginary $\mathbf{Q}[i\sqrt{d}]$ for $d = 1, 2, 3, 7, 11$ and the cyclotomic rings of conductor m such that $\phi(m) < 16$ with $m \notin \{16, 24\}$.

28 Recall that any number field contains $\mathbf{Q}$ so that we are sure to descend to at least $\mathbf{Z}$ at the end of the recursion

Let us go back to the topmost recursive call: when the call ends, we reduced a lattice of rank $2 \times [\mathbf{K}_h : \mathbf{K}_{h-1}]$ corresponding to the descent of the $\mathscr{O}_{\mathbf{K}_h}$-lattice spanned by b_i, b_{i+1} orthogonal projected to the previous vectors of the basis. This reduction allows to find a short vector of the plane $\mathcal{P} = b_i\mathscr{O}_{\mathbf{K}_h} \oplus b_{i+1}\mathscr{O}_{\mathbf{K}_h}$. It then suffices to construct a second vector allowing to complete the first one to form a new basis of $\mathcal{P}$ to pursue the reduction. This completion is done by solving a Bezout-equation with coefficients in $\mathscr{O}_{\mathbf{K}}$.

3.2.4. *Extended Euclidean algorithm in number fields.* A Bezout equation is an equation of the shape:

$$au + bv = 1,$$

with unknown u and v in $\mathbf{Z}$. It has solutions if and only if the integers a and b are *coprime*. The search for solutions is classically done in polynomial time by the so-called *extended Euclidean algorithm*. The generalized version of this equation on number fields has the same shape but with this time $a, b \in \mathscr{O}_{\mathbf{K}_h}$ being algebraic integers. As in the whole generality the ring $\mathscr{O}_{\mathbf{K}}$ is not Euclidean, the extended euclidean algorithm based on subversive Euclidean divisions, can not work directly. However if the tower $\mathbf{Q} \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h$ is not trivial, we can use the structure of this tower to descend the problem on the subfield $\mathbf{K}_{h-1}$,by computing the *relative norm*[29] $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}$ of elements a and b. Then, by recursively calling the resolution algorithm on $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)$ and $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)$, we obtain two algebraic integers μ and ν of $\mathscr{O}_{\mathbf{K}_{h-1}}$ solutions of the equation

$$\mu N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a) + \nu N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b) = 1. \tag{0.2}$$

As a consequence, we can prove that for any elements $\alpha \in \mathscr{O}_{\mathbf{K}_h}$, we have, $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(\alpha) \in \alpha\mathscr{O}_{\mathbf{K}_h}$, so that, $\alpha^{-1}N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(\alpha) \in \mathscr{O}_{\mathbf{K}_h}$. Then we get

$$a \cdot \underbrace{\mu\, a^{-1}N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)}_{:=u\in\mathscr{O}_{\mathbf{K}_h}} + b \cdot \underbrace{\nu\, b^{-1}N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)}_{:=v\in\mathscr{O}_{\mathbf{K}_h}} = 1,$$

as wanted.

It then sufficed to remark that the element u and v found by these algorithms are not necessarily the smallest possible solutions of the Bezout equation. To avoid a blow-up in the size of the coefficients, we need to control the size of the solutions appearing at each descent and lift. This control is made possible by a similar technique to the one allowing to control the size of the coefficients in the reduction algorithm, which is the size-reduction.

Hence the combinations of this lifting with the algorithm we evoked allows a faster reduction on number fields, For a lattice of rank 2 over a cyclotomic field of degree n it suffices of $\mathrm{O}\big(n^2 B \log B\big)$ binary operations, instead of the $\mathrm{O}\big(n^4 B \log B\big)$ required by the algorithm of [128].

29 For a field extension $\mathbf{K}_{h-1} \subset \mathbf{K}_h$, the relative norm $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a) \in \mathbf{K}_{h-1}$ of $a \in \mathbf{K}_h$ is the determinant of the $\mathbf{K}_{h-1}$-linear map $x \mapsto ax$ on $\mathbf{K}_h$, seen as vector space. This form is multiplicative: for any $a, b \in \mathbf{K}_h$, $N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(ab) = N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)N_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)$.

3.3 Using a symplectic structure over number fields

A Euclidean space is a vector space endowed with a positive definite symmetric bilinear form acting on it. Replacing this form by an antisymmetric one yields the notion of *symplectic space*. Lattices embedded in symplectic spaces have additional symmetries that can be exploited to (roughly) halve the cost of the reduction—this is more precisely done by replacing the size-reduction viewed as an integral rounding of the Gram-Schmidt decomposition by a rounding version of the so-called Iwasawa decomposition. We prove that we can define a recursive symplectic structure over a tower of number fields, compatible with the descent between subfields. As a consequence we can halve the running time of the reduction at *each* level of the recursion tree, yielding significant asymptotic speedups on the overall reduction. With this technique, over a cyclotomic field of degree n and sufficiently smooth conductor, we can reduce a rank two module represented as a 2×2 matrix M whose number of bits in the input coefficients is uniformly bounded by $B > n$, in time

$$\tilde{\mathrm{O}}\left(n^{2+\frac{\log(1/2+1/2q)}{\log q}} B\right) + n^{\mathrm{O}(\log\log n)},$$

where q is a prime, and the conductor is a power of q. The first column of the reduced matrix has its coefficients uniformly bounded by $2^{\tilde{\mathrm{O}}(n)} (\mathrm{covol}\, M)^{\frac{1}{2n}}$.

3.4 An application in algebraic number theory: the principal ideal problem

3.4.1. *Ideals.* Since the algebraic integers, $\mathscr{O}_{\mathbf{K}}$ of a number field $\mathbf{K}$ have a ring structure, we can look at their ideals[30]. In particular, if the algebraic integers do not share the property of *unique factorization*[31], its ideals does nonetheless. More precisely any $\mathfrak{a}$ of $\mathscr{O}_{\mathbf{K}}$ decomposes in a product, unique up-to-order, of primes ideals. In this sense, the rings of integers have an ideal arithmetic which is similar to the rational arithmetic of $\mathbf{Z}$. However, it is noticeable that on the contrary to the ideals of $\mathbf{Z}$, ideals in generic rings of integers are not necessarily principal, that is spanned by a unique element. A classical problem in algorithmic number theory is raised by this question: given an ideal of a ring of integer, how difficult is it to compute a generator. In order to ease the readability of the subsequent paragraphs, we denote $\langle a \rangle = a\mathscr{O}_{\mathbf{K}} = \{a \cdot x \,|\, x \in \mathscr{O}_{\mathbf{K}}\}$ the ideal spanned by a in $\mathscr{O}_{\mathbf{K}}$ for any element a.

30 Recall that an ideal $\mathfrak{a}$ is a subgroup for the additive law and stable under the multiplication by $\mathscr{O}_{\mathbf{K}}$: that is for any integer $x \in \mathscr{O}_{\mathbf{K}}$, $x\mathfrak{a} \subseteq \mathfrak{a}$. An ideal is *prime* if the following condition is satisfied: for all $a, b \in \mathscr{O}_{\mathbf{K}}$, $ab \in \mathfrak{a}$ implies that $a \in \mathfrak{a}$ or $b \in \mathfrak{a}$. Its norm is defined as the cardinal of the quotient $\mathscr{O}_{\mathbf{K}}/\mathfrak{a}$. It is a multiplicative form giving a measurement of the size of the ideal: the closest $\mathfrak{a}$ is from $\mathscr{O}_{\mathbf{K}}$, the smaller is its norm.

31 Think of the double factorization $4 = 2 \times 2 = (1 + i\sqrt{3})(1 - i\sqrt{3})$ in the number field $\mathbf{Q}[i\sqrt{3}]$.

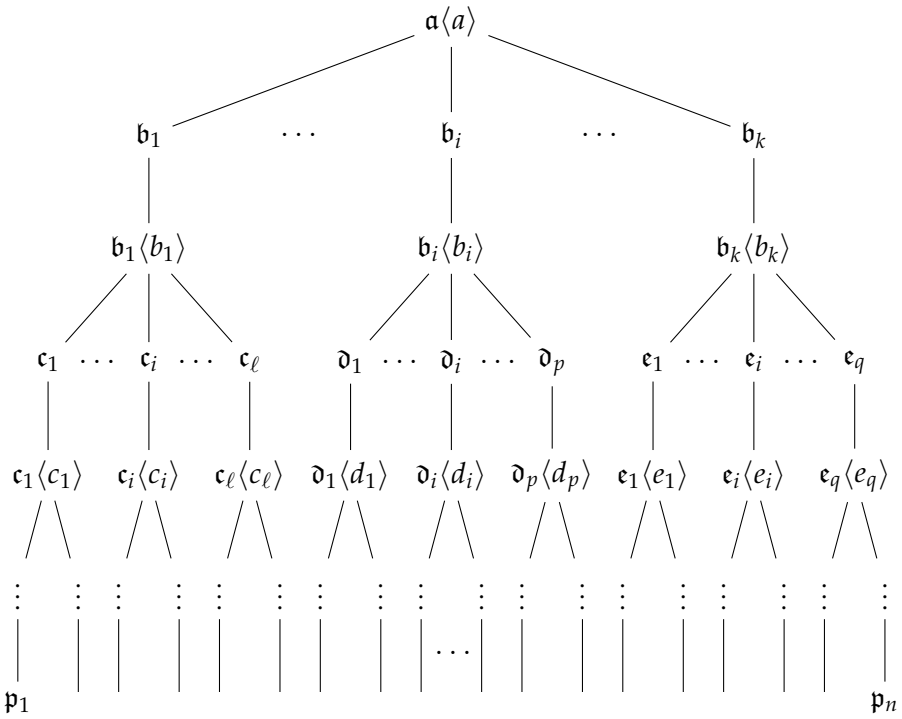


Figure 8: Description of the descent to small norm ideals. The smoothness of the ideals is decreasing along the descent tree. The leaves are all B-smooth.

3.4.2. *Solution on all small primes.* Let us suppose that we know how to resolve this problem for all primes of norm bounded by a certain $B > 0$. Then, by using linear algebra, we can solve the principal ideal problem for all principal ideals which is B-smooth, that is such that all of its prime factors are of norm smaller than B. Hence, to solve the principal ideal problem in all generality, we can reduce the problem for an ideal $\mathfrak{a}$ or arbitrary norm to multiple instances of for B-smooth ideals.

3.4.3. *Descending to B-smooth ideals.* Let $\mathfrak{a}$ be a principal ideal. By the latter remark, we aim at *reducing* the principal ideal problem on $\mathfrak{a}$ to the problem on smooth ideals. To do so we look for an element $a \in \mathfrak{a}$ such that $\mathfrak{a} \cdot \langle a\rangle$ is sufficiently smooth (for a notion of sufficiently smooth that is explicated in Chapter 6). Hence we then factorize this newly constructed ideal in primes factors of norm smaller than the starting ideal. On *each* of these ideals, we can start again and look for a small element to improve their smoothness, and continue this process until getting through the bound B introduced in Paragraph 3.4.2. Figure 8 gives a schematic vision of this phase, called "descent".

Once this descent over, it then suffices to solve the principal ideal problem on each of them, and to propagate the found generators up to the top of the tree to recover a generator of the initial ideal.

3.4.4. *Reduction of an ideal.* In order to complete this brief presentation of the algorithm, we still have to elucidate a point: given an ideal $\mathfrak{a}$, how to construct an element a such that $\mathfrak{a} \cdot \langle a \rangle$ is smoother than the ideal $\mathfrak{a}$? To do so we are going to minimize the norm of $\mathfrak{a} \cdot \langle a \rangle$, which boils down to finding a small principal ideal $\langle a \rangle$ included in $\mathfrak{a}$ and to divide $\mathfrak{a}$ by $\langle a \rangle$. Hence by multiplicativity of the norm, the quotient is of norm smaller than the norm of $\mathfrak{a}$ and the smallest $\langle a \rangle$, the smallest the resulting ideal will be. It then suffices to construct the smallest possible element a in the ideal, that is to say to perform a *reduction of the algebraic lattice* defined by $\mathfrak{a}$.

We actually went back to our leitmotiv: the algorithmic arithmetic problem has been reduced *in fine* to a myriad of instances of search for short vectors in algebraic lattices, as the proof of the two square theorem reduced to the finding of a smallest vector in a well-chosen lattice.

3.5 Cryptographical opening

The XX$^{\text{th}}$ century has been the theater of profound changes in cryptography. Etymologically, the cryptology is the science of secret, but can not really be considered as a science until the end of the 60's, as it was more of a technical field. This science encompasses the cryptography—the art of protecting a secret—and the cryptanalysis—which we could qualify by offensive, consists in the analysis of the schemes produced by the cryptography.

3.5.1. *Mutation of the cryptographical landscape during the XXth century.* From the premises of cryptography, with the famous *scytale* of the Spartans, to the war codes of the First World War, such as the one of the Zimmerman telegram, cryptography was *manual*: all encoding, decoding, and cryptanalysis was based on transpositions of letters, phonemes or words, and was perfumed by hand.

Cryptography was deeply just after the end of the war with the development of the *Enigma* machine in 1919, when a dutch engineer, Hugo Alexander Koch, patent an *electromechanical* machine to encrypt. It is still a *transposition* cipher—each letter is replaced by another one—with the subtlety that the whole substitution change from a letter to another. The encryption is performed by the machine, built on an assiduous system using moving rotors and electric wires. The cryptanalysis was forced to enter this *mechanical* age, as the cryptanalysis of Enigma was introduced by Alan Turing and his so-called *bombs* which allowed the *bruteforce* of the code, that is trying every possible key.

The year 1976 was maybe the most profound change in cryptography: it is the date of publication of the seminal article New Directions in Cryptography [44] of Diffie and Hellman. This article introduced a radically new paradigm to distribute cryptographical keys and solve a fundamental problem of cryptography: the key exchange. This article almost instantly fired up the development of a new kind of cryptographical algorithms, the *asymmet-*

ric schemes. Before that landmark, all ciphers were symmetrical schemes where the same key was used by the sender and the receiver.

We shall now quit this historical aparté and get back to our matter of interest, lattices.

3.6 Lattice based-cryptography

Asymmetrical cryptography requires to find hard problems, in the sense of complexity theory. But beyond the problem itself, finding hard instances of them is a critical matter[32] to instantiate the cryptosystem.

The first method used in cryptography to produce difficult problems on average case has been proposed simultaneously in 1996 by Ajtai on the one hand and by Hoffstein, Pipher et Silverman (NTRUSIGN) [81] on the other hand. Both are based on *Euclidean lattices*. Subsequently, Regev introduced in 2005 the *learning with error problem* (LWE), together with a quantum reduction to a classical lattice problem, the shortest independent vectors problem. Since then lattice-based cryptography has vividly developed and imposes itself as a serious contender for the rise of a "post-quantum" cryptography.

Indeed, it appears that, up to this date, we are not able to design an efficient *quantum* algorithm to solve the hard problems on Euclidean lattices, whereas some of the pillars of asymmetrical cryptography, notably integer factorization and discrete logarithm in finite fields have been undermined by quantum-enhanced search techniques.

In addition to this advantage on post-quantum security, lattices enabled the development of feature-rich primitives such as the homomorphic encryption [62].

Hence, Euclidean lattices constitute a particularly attractive foundation for cryptographical researches, for the offered possibilities as well as for the security promises.

3.7 Lattice reduction as a cryptanalytical toolkit

3.7.1. *Security and reduction* To evaluate the security of lattice-based primitives, it is often necessary to use lattice-reduction algorithms. Indeed, the finding of a short element in a lattice used to built a cryptographical scheme allows generally to reconstruct quite straightforwardly the secret key.

In order to speed-up the lattice-based cryptosystems, it has been suggested, for instance in [82], [114], [105], to switch from arbitrary lattices to ideals and more generally to lattices over number fields. Indeed, the alge-

32 A well-known example is the 3-SAT problem of satisfiability: the instances which are not specifically built to be difficult are actually much easier to solve that what the NP-completeness of the problem would let us think, opening the door to numerous very efficient practical solvers.

braic structure of these lattices allows performing faster computations while minimizing the size of the objects manipulated[33].

Hence the security of such cryptosystems depends on our capacity to reduce algebraic-lattices, justifying the interest of the reduction techniques introduced *supra*.

3.7.2. *Practical examples of cryptanalysis using algebraic lattices.* Chapter 8 presents the cryptanalysis of three lattice-based schemes. The first one is an attack on the homomorphic encryption scheme of Smart and Vercauteren [153]. Its secret key is made of a small element in a cyclotomic field, and the corresponding public key is a $\mathbf{Z}$-basis of the ideal generated by this element. Hence we can attack the scheme with the help of the algorithm for the principal ideal problem, presented in Chapter 6. Remark that finding a generator is not completely sufficient, as we aim for a *short* generator. To do so, we use the reduction of [42] allowing us to find a short generator from an arbitrary one, in polynomial time.

Eventually, we detail a side-channel attack on the BLISS digital signature scheme [49]. A side-channel attack is an attack which aims at discovering and exploiting the flaws in the implementation, in software or hardware, of a cryptographical scheme or protocol. Such an attack does not alter the *theoretical* robustness of a cryptosystem, but solely expose a flaw in its *practical implementation*. An analysis of the power consumption's traces of the BLISS signature scheme allows indeed to estimate the norm over a subfield of the secret key, which is a small element of a cyclotomic field. Using lattice-reduction and the properties of the factorization of ideals, we retrieve the secret key from its norm, that is solving a *norm equation* in a number field.

33 For instance, a lattice coming from an ideal can be represented by two polynomials instead of a full integer matrix.

NOTATIONS AND CONVENTIONS

GENERAL NOTATIONS

The bold capitals $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{R}$ refer as usual to the ring of integers and respectively the field of rational and real. Given a real number x, its integral rounding denoted by $\lfloor x \rceil$ returns its closest. Its fractional part is the excess beyond that number's integer part and denoted by $\{x\}$.

These operators are extended to operate on vectors and matrices by pointwise composition. The complex conjugation of $z \in \mathbf{C}$ is denoted by the usual bar $\bar{z}$. The logarithm functions are used as $\log$ for the binary logarithm and $\ln$ for the natural one.

We say that an integer $n \in \mathbf{Z}$ is *log-smooth* if all the prime factors of n are bounded by $\log(n)$.

MATRIX AND NORMS

For a field $\mathbf{K}$, let us denote by $\mathbf{K}^{d\times d}$ the space of square matrices of size d over $\mathbf{K}$, $\mathrm{Gl}_d(\mathbf{K})$ its group of invertibles. Denote classically the elementary matrices by $T_{i,j}(\lambda)$ and $D_i(\lambda)$ for respectively the transvection (or shear mapping) and the dilatation of parameter λ.

We extend the definition of the product for any pair of matrices (A, B): for every matrix C with compatible size with A and B, we set: $(A, B) \cdot C = (AC, BC)$.

For a vector v (resp. matrix A), we denote by $\|v\|_\infty$ (resp. $\|A\|_{\max}$) its absolute (resp. max) norm, that is the maximum of the absolute value of its coefficients.

For any matrix A we geerically denotes by A_i the i-th column of A. We also adopt the following conventions for submatrix extraction: for any matrix $M = (m_{i,j}) \in \mathbf{K}^{n\times n}$ and $1 \leq a < b \leq n, 1 \leq c < d \leq n$, define the extracted submatrix

$$M[a : b, c : d] = \left(m_{i,j}\right)_{a\leq i\leq b, c\leq j\leq d}.$$

For a family of square matrices $J_1, \ldots, J_\ell$ of respective dimensions $n_1 \times n_1, \ldots, n_\ell \times n_\ell$, denote by $\mathrm{Diag}(J_1, \ldots, J_\ell)$ the block diagonal matrix $(\sum_{i=1}^{\ell} n_1) \times (\sum_{i=1}^{\ell} n_i)$ of diagonal elements the J_i, that is:

$$\begin{pmatrix} & & 0 & 0 & 0 & 0 \\ & J_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & 0 & 0 \\ 0 & 0 & & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & & J_\ell \end{pmatrix}$$

COMPUTATIONAL SETTING

We use the standard model in algorithmic theory, i.e. the word-RAM with unit cost and logarithmic size register (see for instance [119, Section 2.2] for a comprehensive reference). The number of bits in the register is generically denoted by w.

For a non-negative integer d, we set $\omega(d)$ to be the exponent of matrix multiplication of $d \times d$ matrices. If the dimension d is clear from context we might omit it and write simply $\mathrm{O}(d^\omega)$ for this complexity. We can assume that this exponent is not too close to 2, in particular $\omega(d) > 2 + 1/\log(d)$, so that complexities with terms in $\mathrm{O}\big((\omega - 2)^{-1}\big)$ makes sense. Also, we assume that ω is non-increasing.

Part I

GEOMETRY OF NUMBERS WITH A FLAVOR OF NUMBER THEORY

This first part introduces the objects and notions used in the subsequent part of the manuscript. It consists in three chapters. The first one provides a basic introduction to algebraic number theory, with an emphasis on its effective aspects. Then, we move to a short introduction to Minowski's geometry of numbers, viewed through the prism of lattice reduction algorithms. Eventually, we make use of this background to present the theory of lattices over number fields, or so-called Humbert forms. We conclude this third chapter by exposing the difficulties yields when trying to extend the lattice reduction algorithms to this more general context.

1

A BRIEF INTRODUCTION TO ALGEBRAIC NUMBER THEORY

The aim of this chapter is to introduce the algebraic notions used in the subsequent parts of this manuscript. It reviews standard concepts of commutative algebra and algebraic number theory. For a much more detailed and comprehensive reference, we invite the interested reader to refer to the monograph of Jungern Neukirch [127], the more synthetic reference [142] of Pierre Samuel, or the classical Algebraic number theory of Lang [104]. The first part of this chapter is devoted to module theory, with an emphasis on torsion-free modules over Dedekind domains, since all of the rings we encounter in our works are actually be Dedekind domains. Then we invite the reader to a small journey into elementary number fields theory with a final focus on cyclotomic fields.

Since the motto of the present manuscript is to give an algebraic insight on the algorithmic counterpart of a geometry of numbers over number fields, we put emphasis all along this chapter on the algorithmic methods used to manipulate and deal with the introduced objects. For an extensive reference on algorithms in number theory, we let the reader refer to the two books of Henri Cohen, [34] and [35].

1.1 MODULE OVER DEDEKIND DOMAINS

Modules over rings are a fundamental structure in algebra, appears naturally in many branches of mathematics, like algebraic geometry, algebraic topology and of course, number theory. A module is an additive abelian group endowed with a product between elements of the ring and its element, which is distributive over the addition operation of each parameter and is compatible with the ring multiplication. In a broad sense, modules can be thought as a generalization of vector spaces, where the base field is replaced by an arbitrary ring.

Since the work presented in this manuscript only concerns the algebraic part of number theory, all the rings we encounter are actually commutative, and thus we simplify this introduction by sticking to the abelian case, allowing to avoid the distinction between left and right actions of the scalars.

1.1.1 Basic definitions

Let R be a ring. We denote by 0 and 1 its respective additive and multiplicative neutral elements. A R-module $\mathcal{M}$ consists of an abelian group $(\mathcal{M}, +)$

and an operation $\bullet : R \times \mathcal{M} \to \mathcal{M}$ such that for all $r, s \in R$ and $m, n \in \mathcal{M}$, we have:

- $r \cdot (m + n) = r \cdot m + r \cdot n$ *(linearity with vectors)*
- $(r + s) \cdot m = r \cdot m + s \cdot m$ *(linearity with scalars)*
- $(rs) \cdot m = r \cdot (s \cdot m)$ *(associativity)*
- $1 \cdot m = m$ *(action of the neutral element)*

As for vector spaces, the action of the ring on $\mathcal{M}$ is called *scalar multiplication.*

Example. *A central example in this manuscript is given by the class of modules of the form $\mathbf{Z}^n$ for an integer n, which can be seen as the set of n-dimensional vectors with integral coefficients.*

1.1.2 Submodules, morphism

Suppose $\mathcal{M}$ is an R-module and $\mathcal{N}$ is a subgroup of $\mathcal{M}$. Then $\mathcal{N}$ is a submodule (or more explicitly an R-submodule) if for any $n \in \mathcal{N}$ and any $r \in R$, the product $r \cdot n$ is still in $\mathcal{N}$.

If $\mathcal{M}$ and $\mathcal{N}$ are two R-modules, then a map $f : \mathcal{M} \to \mathcal{N}$ is a homomorphism of R-modules (or a R-linear map) if for any $m, n \in \mathcal{M}$ and $r, s \in R$,

$$f(r \cdot m + s \cdot n) = r \cdot f(m) + s \cdot f(n).$$

It is an *isomorphism* if it is also bijective as a map between sets.

As usual for algebraic structures, the kernel—that is, the preimages of 0—of an R-linear map between two R-modules $\mathcal{M}$ and $\mathcal{N}$ is a R-submodule of $\mathcal{M}$.

1.1.3 Direct sums and tensor product over modules

Let us fix a commutative ring R and let $\mathcal{M}$ and $\mathcal{N}$ be two R-modules.

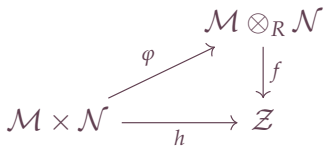
1.1.3.1. *Direct sum.* The cartesian product $\mathcal{M} \times \mathcal{N}$ can be given the structure of R-module by defining the operations componentwise:

- $(m, n) + (m', n') = (m + n, m' + n')$
- $r \cdot (m, n) = (r \cdot m, r \cdot n)$

for any $m, m' \in \mathcal{M}, n, n' \in \mathcal{N}$ and $r \in R$. This module structure is denoted by $\mathcal{M} \oplus \mathcal{N}$, called direct sum of $\mathcal{M}$ and $\mathcal{N}$. This is the smallest module which contains the given modules as submodules.

Example. *Taking two non-negative integers, say n and m, allows us to construct the direct sum $\mathbf{Z}^n \oplus \mathbf{Z}^m$. Following the interpretation given above, an element of $\mathbf{Z}^n \oplus \mathbf{Z}^m$ is a couple consisting of a n-dimensional vector and of a m-dimensional vector with integer coefficients, that is a $(n+m)$-dimensional vector of integers. The addition rule and product rule let us easily verify that the resulting laws are the same as the laws of $\mathbf{Z}^{n+m}$, so that $\mathbf{Z}^n \oplus \mathbf{Z}^m \cong \mathbf{Z}^{n+m}$.*

1.1.3.2. *Tensor products, extension of scalars.* Tensor product of modules is a construction that allows arguments about bilinear maps to be written in terms of linear maps. More precisely it satisfies the following universal property:
given $\mathcal{M}$ and $\mathcal{N}$ two R-modules, their tensor product is the data of an R-module $\mathcal{M} \otimes_R \mathcal{N}$ and a bilinear map $\varphi : \mathcal{M} \times \mathcal{N} \to \mathcal{M} \otimes_R \mathcal{N}$ which have the property that any bilinear map $h : \mathcal{M} \times \mathcal{N} \to \mathcal{Z}$ for an R-module $\mathcal{Z}$ factors uniquely through φ in a R-linear map f, that is making the following diagram commutative:



The construction of the tensor product $\mathcal{M} \otimes_R \mathcal{N}$ takes a quotient of the free abelian group with basis the abstract symbols $m * n$—used here to denote the ordered pair (m, n)—for m in $\mathcal{M}$ and n in $\mathcal{N}$, by the subgroup generated by all elements of the form:

- $-m * (n + n') + m * n + m * n'$
- $-(m + m') * n + m * n + m' * n$
- $(m \cdot r) * n - m * (r \cdot n)$

where m, m' in $\mathcal{M}$, n, n' in $\mathcal{N}$, and r in R. The quotient map takes $m * n = (m, n)$ to the coset containing $m * n$; that is,

$$\mathcal{M} \times \mathcal{N} \to \mathcal{M} \otimes_R \mathcal{N},\ (m, n) \mapsto [m * n].$$

This abelian group has a natural structure of R-module since R is commutative, as $rr'(m \otimes n) = rm \otimes r'n = r'm \otimes rn$, for any $r, r' \in R$ and $m, n \in \mathcal{M} \times \mathcal{N}$.

A useful application of the tensor construction is the so-called "extension of scalar". Let $f : R \to S$ be a homomorphism between two rings, and let $\mathcal{M}$ be a R-module. Consider the tensor product $\mathcal{M}_S = \mathcal{M} \otimes_R S$, where S is considered as a R-module via the map f. Since S is also a module over itself, and the two actions commute, that is $r \cdot (s \cdot s') = (r \cdot s) \cdot s'$ for any $r \in R$, and $s, s' \in S$, the module $\mathcal{M}_S$ inherits the action S. It is $(m \otimes s) \cdot s' = m \otimes ss'$ for any $m \in \mathcal{M}$.

Example. *A particularly interesting example is the completion of a module over an integral domain in a number field by extending the ring into its fraction field. For instance, if one consider a* $\mathbf{Z}$*-module* $\mathcal{M}$*, then its extension* $\mathcal{M} \otimes_{\mathbf{Z}} \mathbf{Q}$ *is a* $\mathbf{Q}$*-vector space containing* $\mathcal{M}$*.*

1.1.4 A small taxonomy of modules

In the following, we briefly introduce and discuss the properties of the modules we are going to encounter. For the sake of simplicity, we now only consider rings which are *integral domains*—i.e. rings where the products of two nonzero elements is not zero—, even though the following constructions can be adapted for any ring. This allows us to always consider the fraction field of any ring used.

1.1.4.1. *Finitely generated modules.* A R-module $\mathcal{M}$ is *finitely generated* if there exists a family of elements $(a_1, a_2, \dots, a_d)$ of $\mathcal{M}$ such that for any $x \in \mathcal{M}$, there exist some $r_1, r_2, \dots, r_d$ in R such that $x = r_1 a_1 + r_2 a_2 + \cdots + r_d a_d$.

The set $(a_1, a_2, \dots, a_d)$ is referred to as a *generating set* for $\mathcal{M}$ in this case. A *rank* notion can be defined on finitely generated modules, which can be seen as a generalization of the dimension in linear algebra. Denote by K the fraction field of R. Formally the *rank* $\operatorname{rk} \mathcal{M}$ of the module $\mathcal{M}$ over the domain R is the dimension of the space $(\mathcal{M} \otimes_R K)$ seen as a vector space over the field K. The effect of this scalar extension is to *kill the torsion* in the module $\mathcal{M}$ in the following sense:

Fact.

$$\mathcal{M} \otimes_R K \cong {}^{\mathcal{M}}\!/_{t(\mathcal{M})} \otimes_R K,$$

where $t(\mathcal{M}) = \{m \in \mathcal{M} \mid \exists r \in R \setminus 0,\, r \cdot m = 0\}$ *is the torsion submodule*[1] *of* $\mathcal{M}$*.*

Proof. Denote by π the canonical projection of $\mathcal{M}$ over the quotient ${}^{\mathcal{M}}\!/_{t(\mathcal{M})}$. Let us define:

$$F : \left| \begin{array}{ccc} \mathcal{M} \times K & \longrightarrow & {}^{\mathcal{M}}\!/_{t(\mathcal{M})} \otimes_R K \\ (m, k) & \longmapsto & (\pi(m) \otimes k) \end{array} \right. .$$

This map is clearly bilinear, so there exists a R-linear map $f : \mathcal{M} \otimes_R K \longrightarrow {}^{\mathcal{M}}\!/_{t(\mathcal{M})} \otimes_R K$ for which $f(m \otimes k) = \pi(m) \otimes k$, that is, lifting the projection π. Conversely, remark that the tensor mapping $\otimes : \mathcal{M} \times K \longrightarrow \mathcal{M} \otimes_R K$ vanishes over $t(\mathcal{M}) \times K)$. As such we can factor this map into a bilinear mapping:

$$G : \left| \begin{array}{ccc} {}^{\mathcal{M}}\!/_{t(\mathcal{M})} \times K & \longrightarrow & \mathcal{M} \otimes_R K \\ (\pi(m), k) & \longmapsto & (m \otimes k) \end{array} \right. .$$

1 We have used implicitly in this definition of the torsion that R is an integral domain.

This map is well defined since the pure tensor $m \otimes k$ only depends on m through its class in the quotient $\mathcal{M}/t(\mathcal{M})$. Hence, there exists a R-linear map $g : \mathcal{M}/t(\mathcal{M}) \otimes_R K \longrightarrow \mathcal{M} \otimes_R K$ such that $g(\pi(m) \otimes k) = m \otimes k$. As $f \circ g$ and $g \circ f$ fixe all pure tensors and are linear, the let invariant all tensors and thus f and g are inverse isomorphisms. ■

Remark that the rank notion is compatible with the direct sum and tensor product in the sense that for any R-modules $\mathcal{M}$ and $\mathcal{N}$ we have: $\operatorname{rk}(\mathcal{M} \oplus \mathcal{N}) = \operatorname{rk}\mathcal{M} + \operatorname{rk}\mathcal{N}$ and $\operatorname{rk}(\mathcal{M} \otimes \mathcal{N}) = \operatorname{rk}\mathcal{M} \times \operatorname{rk}\mathcal{N}$.

Example. *Finitely generated modules over $\mathbf{Z}$ are by definition the finitely generated abelian groups. The structure theorem (see for instance [154]) asserts that such a module $\mathcal{M}$ is isomorphic to $\mathbf{Z}^n \oplus \mathbf{Z}/q_1\mathbf{Z} \oplus \cdots \oplus \mathbf{Z}/q_t\mathbf{Z}$ for a non negative integer n and $q_1, \ldots, q_t$ are powers of (not necessarily distinct) prime numbers. The rank of $\mathcal{M}$ is n as $\mathcal{M} \otimes_{\mathbf{Z}} \mathbf{Q} \cong \mathbf{Z}^n \otimes_{\mathbf{Z}} \mathbf{Q} \cong \mathbf{Q}^n$.*

1.1.4.2. *Free modules.* For an integral domain R and an R-module $\mathcal{M}$, a subset $\mathcal{B} \subseteq M$ is a basis for $\mathcal{M}$ if:

- $\mathcal{B}$ is a generating set for $\mathcal{M}$.
- $\mathcal{B}$ is linearly independent, that is, $r_1e_1 + r_2e_2 + \cdots + r_de_d = 0$ for $e_1, e_2, \ldots, e_d$ distinct elements of $\mathcal{B}$ and $r_1, \ldots, r_d \in R$ implies that $r_1 = \cdots = r_d = 0$.

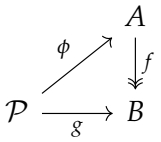
A free module is a module with a basis. A simple, yet central, example of free module is given by quotients of polynomial rings:

Example. *Let $A[X]$ be a polynomial ring over a commutative ring A and f a monic polynomial of degree d in $A[x]$. Define the quotient module $B = A[X]/(f)$ and set ξ to be the image of X in B. Then B contains A as a subring and is free as an A-module with a basis $1, \xi, \ldots, \xi^{d-1}$, called its* power basis.

The notion of freeness is also linked to the rank introduced in Paragraph 1.1.4.1: let $\mathcal{M}$ be a finitely generated module over R, then its rank is also equal to the rank of any maximal free submodule of $\mathcal{M}$.

1.1.4.3. *Projective modules.* The notion of projective modules is a slight generalization of free modules. Indeed, every free module is projective but the contrary does not hold.

A module $\mathcal{P}$ is projective if and only if there is another module $\mathcal{Q}$ such that the direct sum $\mathcal{P} \oplus \mathcal{Q}$ is free. Equivalently, this means that for any surjective R-module morphism f from some modules A to B, and any morphism $g : \mathcal{P} \to B$, there exists a lift morphism ϕ which makes the following diagram commutative:



An example of a finitely generated projective module which is not free is given and studied in Chapter 3.

1.1.5 Modules over Dedekind domains

1.1.5.1. *Ideal and fractional ideals.*

Definition 1.1.1. *An* ideal, *or* integral ideal, *of R is a non-zero R-submodule of R. More generally a* fractional ideal $\mathfrak{a}$ *of R is a nonzero R-submodule of the fraction field K of R for which there exists a nonzero x in K such that $x\mathfrak{a} \subset R$.*

1.1.5.2. *Multiplication.* The product of two fractional ideals $\mathfrak{a}$ and $\mathfrak{b}$ is defined as follows

$$\mathfrak{ab} := \{a_1b_1 + \cdots + a_mb_m \mid a_i \in \mathfrak{a} \text{ and } b_i \in \mathfrak{b}, i \in \{1, \ldots, m\}; \forall m \in \mathbf{N}\},$$

i.e., the product is the fractional ideal generated by *all products ab* with $a \in \mathfrak{a}$ and $b \in \mathfrak{b}$. Clearly, R is the neutral element of this multiplication, giving the set of fractional ideals a monoid structure.

1.1.5.3. *Prime ideals.*

Definition 1.1.2 (Prime ideal)**.** *Let R be a ring. An* prime *ideal $\mathfrak{p} \subseteq R$ is an ideal such that $R/\mathfrak{p}$ is an integral domain, that is, for all $x, y \in R$, $xy \in \mathfrak{p}$ implies $x \in \mathfrak{p}$ or $y \in \mathfrak{p}$.*

Lemma 1.1.1. *Let $\mathfrak{p}$ be a prime ideal in a ring R. Then for $\mathfrak{a}, \mathfrak{b} \subseteq R$ ideals, $\mathfrak{ab} \subseteq \mathfrak{p}$ implies $\mathfrak{a} \subseteq \mathfrak{p}$ or $\mathfrak{b} \subseteq \mathfrak{p}$.*

Proof. If it was not the case, then there would be some $a \in \mathfrak{a} \setminus \mathfrak{p}$ and $b \in \mathfrak{b} \setminus \mathfrak{p}$. Then $ab \in \mathfrak{ab} \subseteq \mathfrak{p}$. But then $a \in \mathfrak{p}$ or $b \in \mathfrak{p}$. Contradiction. ∎

1.1.5.4. *Dedekind domains.* We now specialize these notions in the context of Dedekind domains.

Definition 1.1.3 (Dedekind domain)**.** *A ring R, with fraction field K, is said to be a Dedekind domain when it satisfies the following properties:*

- *R is integrally closed: every element of K which is annihilated by a monic polynomial of $R[X]$, lies in R.*
- *R is a Noetherian domain: there is no infinite strictly ascending sequence of ideals.*
- *Every nonzero prime ideal is maximal.*

In all of the following let us fix a Dedekind domain R and set K to be its fraction field.

1.1.5.5. *Properties of ideals over Dedekind domains.*

Definition 1.1.4 (Invertible fractional ideal)**.** *A fractional* invertible *ideal* $\mathfrak{a}$ *is an ideal such that there exists a fractional ideal* $\mathfrak{b}$ *such that* $\mathfrak{ab} = R = \langle 1 \rangle$.

The inverse of fractional ideals are very easy to describe:

Proposition 1.1.1. *In a Dedekind domain* R *of fraction field* K*, every non-zero fractional ideal is invertible. The inverse of* $\mathfrak{a}$ *is* $\{x \in K : x\mathfrak{a} \subseteq R\}$.

Before proving this proposition we need a technical lemma on the denominators of integral ideals.

Lemma 1.1.2. *For any proper integral ideal* $\mathfrak{a}$*, there is some* $\gamma \in K \setminus R$ *for which* $\gamma\mathfrak{a} \subseteq R$.

Proof. Take any nonzero $a \in \mathfrak{a}$. Then aR contains a product of prime ideals, say $aR \supseteq \mathfrak{p}_1\mathfrak{p}_2 \cdots \mathfrak{p}_n$ with n minimal (i.e. $\mathfrak{a}$ does not contain a product of $n-1$ prime ideals). Indeed, assume that this is not the case. Let $\mathcal{S}$ be the collection of integral ideals of R not containing a product of prime ideals, so $\mathcal{S}$ is nonempty and $R \notin \mathcal{S}$. As R is noetherian, $\mathcal{S}$ must have a maximal element, say $\mathfrak{s}$. Clearly $\mathfrak{s}$ cannot be prime (otherwise it would contain a prime: itself), so there must be $x, y \in R$ with $xy \in \mathfrak{s}$ but $x, y \notin \mathfrak{s}$. But then $\mathfrak{s} \subsetneq \mathfrak{s} + (x), \mathfrak{s} + (y)$, and so $\mathfrak{s} + (x)$ and $\mathfrak{s} + (y)$ contain products of prime ideals. But then $(\mathfrak{s} + (x))(\mathfrak{s} + (y)) = \mathfrak{s} + (xy) \subseteq \mathfrak{s}$ also contains a product of prime ideals, contradicting the choice of $\mathfrak{s}$.

As $R \not\subseteq aR$, $n \geq 1$. As $\mathfrak{a}$ is a proper ideal, it must be contained in some maximal ideal, $\mathfrak{p}$. Since maximal ideals are prime in commutative rings, $\mathfrak{p}$ is prime. But now $\mathfrak{p}_1\mathfrak{p}_2 \cdots \mathfrak{p}_n \subseteq \mathfrak{p}$. Thus as $\mathfrak{p}$ is prime, $\mathfrak{p}_i \subseteq \mathfrak{p}$ for some i (if $\mathfrak{p}$ is prime and A, B are ideals with $AB \subseteq \mathfrak{p}$ then either $A \subseteq$ or $B \subseteq \mathfrak{p}$). But as R is a Dedekind domain, $\mathfrak{p}_i$ must be maximal, so $\mathfrak{p} = \mathfrak{p}_i$. Now assume without loss of generality that $i = n$. By the minimality of n, $(\mathfrak{p}_1 \cdots \mathfrak{p}_{n-1}) \not\subseteq aR$. Take any $b \in \mathfrak{p}_1 \cdots \mathfrak{p}_{n-1} \setminus aR$ and let $\gamma = b/a \in K$. We claim that this is the desired γ.

First if $\gamma \in R$ then $b = \gamma a \in aR$, which is a contradiction, so $\gamma \notin R$. Now for any $x \in \mathfrak{a}$, $bx \in \mathfrak{p}_1 \cdots \mathfrak{p}_{n-1}\mathfrak{a} \subseteq \mathfrak{p}_1 \cdots \mathfrak{p}_n \subseteq aR$, and so $bx = ar$ for some $r \in R$. But now $\gamma x = \frac{bx}{a} = r \in R$, and so $\gamma\mathfrak{a} \subseteq R$, as required. ∎

Proof of Proposition 1.1.1. Note that for any $n \in R$ non-zero, we know $\mathfrak{a}$ is invertible if and only if $n\mathfrak{a}$ is invertible. If the annonced result is false, there is an integral ideal $\mathfrak{a} \subseteq R$ which is not invertible. Moreover, as R is Noetherian, we can assume $\mathfrak{a}$ is maximal with this property, i.e. if $\mathfrak{a} \subset \mathfrak{a}' \subset R$, then $\mathfrak{a}'$ is invertible.

Let $\mathfrak{b} = \{x \in K : x\mathfrak{a} \subseteq R\}$, which is a fractional ideal. We clearly have $R \subseteq \mathfrak{b}$, and by Lemma 1.1.2, we know this inclusion is strict.

As $R \subseteq \mathfrak{b}$, we know $\mathfrak{a} \subset \mathfrak{ab}$. Again, this inclusion is strict—if $\mathfrak{ab} = \mathfrak{a}$, then for all $x \in \mathfrak{b}$, we have $x\mathfrak{a} \subseteq \mathfrak{a}$. Hence $R[x]\mathfrak{a} \subseteq \mathfrak{a}$, so that $R[x]$ is a fractional ideal. As such it is finitely generated as R-module as R is Noetherian and x

is therfore annihilated by a polynomial with coefficients in R. By intergral closedness, $x \in R$, but remark that we cannot have $\mathfrak{b} \subseteq R$.

So $\mathfrak{a} \subsetneq \mathfrak{ab}$. By assumption, we know that $\mathfrak{ab} \subseteq R$, and since $\mathfrak{a}$ is not invertible, the inclusion is strict. But then by definition of $\mathfrak{a}$, $\mathfrak{ab}$ is invertible, which implies $\mathfrak{a}$ is invertible (if $\mathfrak{c}$ is an inverse of $\mathfrak{ab}$, then $\mathfrak{bc}$ is an inverse of $\mathfrak{a}$). This is a contradiction. So all fractional ideals must be invertible.

Finally, we have to show that the formula for the inverse actually holds. We write

$$\mathfrak{c} = \{x \in K : x\mathfrak{a} \subseteq R\}.$$

Then by definition, we know $\mathfrak{a}^{-1} \subseteq \mathfrak{c}$. Hence $R = \mathfrak{a}\mathfrak{a}^{-1} \subseteq \mathfrak{ac} \subseteq R$. Therefore, we must have $\mathfrak{ac} = R$, i.e. $\mathfrak{c} = \mathfrak{a}^{-1}$. ∎

Proposition 1.1.2. *Every fractional ideal $\mathfrak{a}$ of a Dedekind R is a finitely generated module, i.e., it can be expressed as $\alpha_1 R + \cdots + \alpha_k R$, for some integer k with the (α_i) belongings to R.*

Proof. Take $\mathfrak{a}$ a fractional ideal. We take $x \in K \setminus \{0\}$ such that $x\mathfrak{a} \subseteq R$, which is an ideal of R. Since this ring is Noetherian, $x\mathfrak{a} \subseteq R$ is finitely generated—otherwise, we could construct inductively an infinite strictly increasing chain by adding a new generator at each step—and hence $\mathfrak{a}$ is finitely generated as then $x\mathfrak{a} \cong \mathfrak{a}$ as R-module since x invertible in K. ∎

1.1.5.6. *Divisibility of ideal.* Similarly to the divisibility notion over the rational integers we can define:

Definition 1.1.5 (Divisibility of ideals)**.** *Let $\mathfrak{a}$ and $\mathfrak{b}$ two ideals: we say that* $\mathfrak{a}$ divides $\mathfrak{b}$*, and write $\mathfrak{a} \mid \mathfrak{b}$, if there exists some ideal $\mathfrak{c}$ such that $\mathfrak{ac} = \mathfrak{b}$.*

Divisibility defines a relation which is, in fact, the inclusion relation:

Corollary 1.1.1. *Let $\mathfrak{a}, \mathfrak{b}, \mathfrak{c} \subseteq R$ be ideals, $\mathfrak{c} \neq 0$. Then*

1. *$\mathfrak{b} \subseteq \mathfrak{a}$ if and only if $\mathfrak{bc} \subseteq \mathfrak{ac}$*
2. *$\mathfrak{a} \mid \mathfrak{b}$ if and only if $\mathfrak{ac} \mid \mathfrak{bc}$*
3. *$\mathfrak{a} \mid \mathfrak{b}$ if and only if $\mathfrak{b} \subseteq \mathfrak{a}$.*

Proof.

1. $(\Rightarrow)$ is clear, and $(\Leftarrow)$ is obtained by multiplying by $\mathfrak{c}^{-1}$.
2. $(\Rightarrow)$ is clear, and $(\Leftarrow)$ is obtained by multiplying by $\mathfrak{c}^{-1}$.
3. $(\Rightarrow)$ is clear. For the other direction, we notice that the result is easy if $\mathfrak{a} = \langle\alpha\rangle$ is principal. Indeed, if $\mathfrak{b} = \langle\beta_1, \ldots, \beta_r\rangle$, then $\mathfrak{b} \subseteq \langle\alpha\rangle$ means there are some $\beta'_a, \ldots, \beta'_r \in R$ such that $\beta_i = \beta'_i\alpha$. But this means:
$$\langle\beta_a, \ldots, \beta_r\rangle = \langle\beta'_1, \ldots, \beta'_r\rangle\langle\alpha\rangle,$$
So $\langle\alpha\rangle \mid \mathfrak{b}$.

   Now suppose we have $\mathfrak{b} \subseteq \mathfrak{a}$. By the proposition, there exists an ideal $\mathfrak{c} \subseteq R$ such that $\mathfrak{ac} = \langle\alpha\rangle$ is principal with $\alpha \in R, \alpha \neq 0$. Then

- $\mathfrak{b} \subseteq \mathfrak{a}$ if and only if $\mathfrak{bc} \subseteq \langle \alpha \rangle$ by (i); and
- $\mathfrak{a} \mid \mathfrak{b}$ if and only if $\langle \alpha \rangle \mid \mathfrak{bc}$ by (ii).

This concludes the proof. ∎

Informally this last lemma states that inclusion of ideals is equivalent to divisibility, this in substance equivalent to saying that "prime ideals are primes of the ideal arithmetic".

1.1.5.7. *Unique factorization.* Eventually, we will prove that every ideal is a product of prime ideals and that this factorization is unique, like for the rational integer arithmetic.

Theorem 1.1.1. *Let $\mathfrak{a} \subseteq R$ be an ideal, $\mathfrak{a} \neq 0$. Then $\mathfrak{a}$ can be written uniquely (up-to-order) as a product of prime ideals.*

Proof. Let us start by the existence. Suppose that $\mathfrak{a}$ is not a prime. Then it is is not maximal, as maximal ideals are clearly primes by definition of primality. Hence, there is some $\mathfrak{b} \supsetneq \mathfrak{a}$ with $\mathfrak{b} \subseteq R$. Hence $\mathfrak{b} \mid \mathfrak{a}$, i.e. there is some $\mathfrak{c} \subseteq R$ with $\mathfrak{a} = \mathfrak{bc}$, and $\mathfrak{c} \supseteq \mathfrak{a}$. We can continue factoring this way, and the process stops eventually, or else we could construct an infinite chain of strictly ascending ideals.

We now prove the uniqueness. We have shown $\mathfrak{p} \mid \mathfrak{ab}$ implies $\mathfrak{p} \mid \mathfrak{a}$ or $\mathfrak{p} \mid \mathfrak{b}$. So if $\mathfrak{p}_1 \cdots \mathfrak{p}_r = \mathfrak{a}_1 \cdots \mathfrak{a}_s$, with $\mathfrak{p}_i, \mathfrak{a}_j$ prime, then we know $\mathfrak{p}_1 \mid \mathfrak{a}_1 \cdots \mathfrak{a}_s$, which implies $\mathfrak{p}_1 \mid \mathfrak{a}_i$ for some i, and without loss of generality we can assume that $i = 1$. So $\mathfrak{a}_1 \subseteq \mathfrak{p}_1$. But $\mathfrak{a}_1$ is prime and hence maximal. So $\mathfrak{p}_1 = \mathfrak{a}_1$. Multiplying the equation $\mathfrak{p}_1 \cdots \mathfrak{p}_r = \mathfrak{a}_1 \cdots \mathfrak{a}_s$ by $\mathfrak{p}_1^{-1}$, and we get $\mathfrak{p}_2 \cdots \mathfrak{p}_r = \mathfrak{a}_2 \cdots \mathfrak{a}_s$. We conclude by a finite induction. ∎

Corollary 1.1.2. *The non-zero fractional ideals form a group under multiplication. We denote this I_K. This is a free abelian group generated by the prime ideals, i.e. any fractional ideal $\mathfrak{a}$ can be written uniquely as $\mathfrak{p}_1^{a_1} \cdots \mathfrak{p}_r^{a_r}$, with $\mathfrak{p}_i$ distinct prime ideals and $a_i \in \mathbf{Z}$. For any i, the integer a_i is called the $\mathfrak{p}_i$-adic valuation of the ideal $\mathfrak{a}$, with the convention to be set at zero for any ideal not appearing in the prime decomposition.*

Moreover, if $\mathfrak{a}$ is an integral ideal, i.e. $\mathfrak{a} \subseteq R$, then $a_1, \ldots, a_r \geq 0$.

Proof. We already have unique factorization of non-fractional ideals. Now take any fractional ideal, and write it as $\mathfrak{a} = \mathfrak{ab}^{-1}$, with $\mathfrak{a}, \mathfrak{b} \in R$ (e.g. take $\mathfrak{b} = \langle x \rangle$ for some x in the fraction field of R), and the result follows directly by simple computation of the exponents. ∎

With this factorization, we can be far more precise on the number of elements required to generate an ideal in a Dedekind domain: for R being a Dedekind domain, then any fractional ideal $\mathfrak{a}$ of R can be generated by two elements. We prove now an ever stronger result:

Proposition 1.1.3. *Let R be a Dedekind domain, then for any fractional ideal $\mathfrak{a}$ of R and $a \in \mathfrak{a}$, there exists $b \in \mathfrak{a}$ so that $\mathfrak{a}$ is generated by a and b.*

Proof. Let $\mathfrak{a}$ and take any $a \in \mathfrak{a}$. Then let factorize the principal ideal aR in $\prod_{i=1}^{r} \mathfrak{p}_i^{a_i}$. Therefore, we have $\mathfrak{a} = \prod_{i=1}^{r} \mathfrak{p}_i^{e_i}$ for exponents $e_i \leq a_i$ as $aR \subset \mathfrak{a}$. Suppose that we can find $b \in R$ such that the $\mathfrak{p}_i$-adic valuation of b is e_i for each $1 \leq i \leq r$. Then $\mathfrak{a}$ divides bR and as such $b \in \mathfrak{a}$. Further, if we set $\mathfrak{a}'$ to be the ideal generated by a and b, then for any prime $\mathfrak{p}$ dividing aR, the $\mathfrak{p}-$adic valuation of $\mathfrak{a}'$ is exactly $\min(a_i, e_i) = e_i$ and for any prone $\mathfrak{q}$ which does not divide aR does not divide $\mathfrak{a}'$. Hence $\mathfrak{a} = \mathfrak{a}'$.

Let us prove that such a b always exists. Set $I = \prod_{i=1}^{r} \mathfrak{p}_i^{e_i+1}$, and $\mathfrak{a}_i = I \cdot \mathfrak{p}_i^{-e_i-1}$, which is of course an integral ideal of R. Then $\mathfrak{a}_1 + \cdots + \mathfrak{a}_r = R$, otherwise the whole sum would be divided by some common prime ideal. Hence there exists a family of $x_i \in \mathfrak{a}_i$ such that $\sum_{i=1}^{r} x_i = 1$. Then, for any set of elements $b_i \in \mathfrak{p}^{e_i} \setminus \mathfrak{p}^{e_i-1}$, the element $\sum_{i=1}^{r} x_i b_i$ satisfies the condition on the valuations. ∎

1.1.5.8. *On finitely generated modules over a Dedekind domain.* We conclude this review on modules by a structure theorem on modules over Dedekind domain, which turns out to be very *rigid*: these modules decompose as direct sums of fractional ideals. Since we are interested by $\mathbf{Z}$-modules in the context of lattices, we start by discussing the structure of projective $\mathbf{Z}$-modules, which will forge the intuition for the general case. Let $\mathcal{M}$ be a projective $\mathbf{Z}$-module. Hence $\mathcal{M}$ is a direct factor of $\mathbf{Z}^n$ for a certain n. As a subgroup of the abelian group $\mathbf{Z}^n$, there exists $r \leq n$ so that $\mathcal{M} \cong \mathbf{Z}^r$, proving that $\mathcal{M}$ is free that is

$$\mathcal{M} = \mathbf{Z} \oplus \cdots \oplus \mathbf{Z}.$$

This boils down to split $\mathcal{M}$ as a direct sum of ideals of the ring $\mathbf{Z}$, which are isomorphic to $\mathbf{Z}$ itself by principality.

The general case is more complicated, as all ideals are not necessarily principal. However, it appears that any projective module is still a direct sum of fractional ideals (in the case of $\mathbf{Z}$, since every ideal is isomorphic to $\mathbf{Z}$ itself, it recovers the previous isomorphism). This is Steinitz' theorem:

Theorem 1.1.2 (Steinitz' classification)**.** *Let R be a Dedekind domain and let $\mathcal{P}$ be a finitely generated projective R-module of rank d. Then*

$$\mathcal{P} \cong \mathfrak{a}_1 \oplus \cdots \oplus \mathfrak{a}_d$$

for $\mathfrak{a}_i$ a family of non-zero fractional ideals. Besides, the class of $[\mathfrak{a}_1 \cdots \mathfrak{a}_d]$, called the Steinitz class, in the class group of R, that is in the quotient of the group of fractional ideals by the principal ideal, is uniquely determined by $\mathcal{P}$.

Proof. Refer for instance to [127]. ∎

Definition 1.1.6 (Pseudo-basis)**.** *Let R be a Dedekind domain and let $\mathcal{P}$ be a finitely generated projective R-module of rank d. Then a family*

$$((v_1, \mathfrak{a}_1), \ldots, (v_d, \mathfrak{a}_d))$$

such that:

$$\mathcal{P} \cong v_1\mathfrak{a}_1 \oplus \cdots \oplus v_d\mathfrak{a}_d,$$

is called a pseudo-basis of $\mathcal{P}$.

1.2 A BIRD'S EYE VIEW ON ALGEBRAIC NUMBER THEORY

In this section, we review the notions of algebraic number theory that will be used in the further developments of this manuscript. A far more complete introduction to this topic can be found in the monograph *Algebraic Number theory* of Neukirch ([127]) for instance.

1.2.1 Number fields

1.2.1.1. *Extensions.*

Definition 1.2.1. *Let* $\mathbf{K}$ *be a field. A* field extension *of* $\mathbf{K}$ *is a field* $\mathbf{L}$ *that contains* $\mathbf{K}$*. It is denoted by* $\mathbf{L}/\mathbf{K}$*.*

If $\mathbf{L}/\mathbf{K}$ is a field extension, then the multiplication of $\mathbf{K}$ on $\mathbf{L}$ defines a $\mathbf{K}$-vector space structure on $\mathbf{L}$. The *degree* $[\mathbf{L} : \mathbf{K}]$ of $\mathbf{L}/\mathbf{K}$ is then the dimension $\dim_{\mathbf{K}}(\mathbf{L})$.

$\mathbf{L}/\mathbf{K}$ is called *finite* if $[\mathbf{L} : \mathbf{K}] < \infty$. If $\mathbf{L}/\mathbf{K}$ is a field extension and $\alpha \in \mathbf{L}$, denote by $\mathbf{K}(\alpha)$ is the subfield of $\mathbf{L}$ generated by $\mathbf{K}$ and α. Explicitly we have:

$$\mathbf{K}(\alpha) = \left\{ \frac{f(\alpha)}{g(\alpha)} \mid f, g \in \mathbf{K}[X], g(\alpha) \neq 0 \right\}.$$

Such field extensions generated by one element are called *simple*.

Any element α of a finite extension $\mathbf{L}/\mathbf{K}$ is annihilated by a polynomial of $\mathbf{K}[X]$ since the extension $\mathbf{K}(\alpha)$ is necessarily finite. The set of all polynomials vanishing at α is an ideal of $\mathbf{K}[X]$ and is, therefore, principal since $\mathbf{K}$ is a field. The unique monic generator ρ_α of this ideal is called the *minimal polynomial* of α. In this case, we have an isomorphism given by identifying α and the image of X in the quotient:

$$\mathbf{K}[X]/(\rho_\alpha) \cong \mathbf{K}(\alpha).$$

1.2.1.2. *Number fields.*

Definition 1.2.2. *A number field* $\mathbf{L}$ *is a finite extension of the rational field* $\mathbf{Q}$*.*

Number fields are simple extension, as we can always find a so-called *primitive element* generating it:

Theorem 1.2.1 (Primitive element theorem)**.** *Let* $\mathbf{L}/\mathbf{Q}$ *be a number field. Then there exists an* $\alpha \in \mathbf{L}$ *such that* $\mathbf{Q}(\alpha) = \mathbf{L}$*.*

Proof. A classical proof of this theorem can be found in [127]. ∎

Example. *We have* $\mathbf{Q}(\sqrt{2}, \sqrt{3}) = \mathbf{Q}(\sqrt{2} + \sqrt{3})$*: the adjunction of two different algebraic integers to* $\mathbf{Q}$ *can be treated by directly studying the primitive element* $\sqrt{2} + \sqrt{3}$*.*

Remark. *By the primitive element theorem, we can represent a number field by the polynomial quotient* $\mathbf{L} \cong \mathbf{Q}[X]/(P)$ *for a polynomial* P *of degree* n *with integral coefficients. Hence we can handle the elements of* $\mathbf{L}$ *by polynomials modulo* (P)*. In practice, this implies that if* α, β *are represented as two polynomials of degree* $n = [\mathbf{L} : \mathbf{Q}]$ *and of coefficients of bitsize* B*, the complexity of computing* $\alpha\beta$ *is* $O(nB + n \log n)$

1.2.2 Algebricity

1.2.2.1. *Ring of integers.* Let $\mathbf{L}/\mathbf{K}$ be a field extension, $\alpha \in \mathbf{L}$. We say that α is *algebraic* over $\mathbf{K}$ if there exists a non-zero polynomial $f \in \mathbf{K}[X]$ with $f(\alpha) = 0$

Definition 1.2.3 (Algebraic integer)**.** *Let* $\mathbf{L}$ *be a number field. An* algebraic integer *is an* $\alpha \in \mathbf{L}$ *such that there is some monic* $f \in \mathbf{Z}[X]$ *vanishing at* α*. We write* $\mathcal{O}_\mathbf{L}$ *for the set of algebraic integers in* $\mathbf{L}$*.*

This definition generalizes the notion of the rational integers $\mathbf{Z}$ as we have $\mathcal{O}_\mathbf{Q} = \mathbf{Z}$, i.e. $\alpha \in \mathbf{Q}$ is an algebraic integer if and only if $\alpha \in \mathbf{Z}$. Further, in this generalization, $\mathcal{O}_\mathbf{L}$ is a subring of $\mathbf{L}$, as $\mathbf{Z}$ is a subring of $\mathbf{Q}$. In addition $\mathbf{L}$ is the fraction field of $\mathcal{O}_\mathbf{L}$.

Definition 1.2.4. *Let* $\mathbf{L}$ *be a number field. A* unit *of* $\mathbf{L}$ *is an element* $x \in \mathcal{O}_\mathbf{L}$ *such that* $x^{-1}1$ *still lies in* $\mathcal{O}_\mathbf{L}$*. The unit group or units of* L^2 *is the ring:*

$$\mathcal{O}_\mathbf{L}^\times = \{x \in \mathcal{O}_\mathbf{L} : x^{-1} \in \mathcal{O}_\mathbf{L}\}$$

1.2.2.2. *Integral bases.*

Definition 1.2.5 (Integral basis)**.** *Let* $\mathbf{L}/\mathbf{Q}$ *be a number field. Then a basis* $\alpha_1, \ldots, \alpha_n$ *of* $\mathbf{L}$ *is an* integral basis *iff*

$$\mathcal{O}_\mathbf{L} = \left\{ \sum_{i=1}^{n} m_i \alpha_i : m_i \in \mathbf{Z} \right\} = \bigoplus_{i=1}^{n} \mathbf{Z}\alpha_i.$$

It is simultaneously a basis for $\mathbf{L}$ *over* $\mathbf{Q}$ *and* $\mathcal{O}_\mathbf{L}$ *over* $\mathbf{Z}$*.*

Example. *Consider* $\mathbf{Q}(\sqrt{d})$ *with* d *square-free,* $d \neq 0, 1$*. If* $d \equiv 1 \pmod 4$*, then* $(1, \frac{1}{2}(1 + \sqrt{d}))$ *is an integral basis. Otherwise, if* $d \equiv 2, 3 \pmod 4$*, then* $(1, \sqrt{d})$ *is an integral basis.*

An integral basis always exists, implying that the ring of integers of a number fields always have a natural structure of *free* $\mathbf{Z}$-module.

Theorem 1.2.2. *Let* $\mathbf{L}/\mathbf{Q}$ *be a number field. Then there exists an integral basis for* $\mathcal{O}_\mathbf{L}$*. In particular,* $\mathcal{O}_\mathbf{L} \cong \mathbf{Z}^n$ *with* $n = [\mathbf{L} : \mathbf{Q}]$*.*

We defer the proof of this theorem to Section 1.3.3.

Remark. *While it is always true that we can find an integral basis and that a power basis always exists for a number field* $\mathbf{L}$*, it is noticeable that an* integral power basis *does not always exist, so that* $\mathscr{O}_{\mathbf{L}}$ *is not necessarily isomorphic to* $\mathbf{Z}[\alpha]$ *for some* $\alpha \in \mathbf{L}$*. An example of such field is not completely trivial and one of the simplest comes back to Dedekind who provided the field* $\mathbf{Q}(\theta)$ *for* θ *being a root of* $X^3 - X^2 - 2X - 8$*, for which* $\mathscr{O}_{\mathbf{Q}(\theta)} = \mathbf{Z} \oplus \theta\mathbf{Z} \oplus (\theta + \theta^2)/2\mathbf{Z}$*. Indeed, if we suppose that* $\mathscr{O}_{\mathbf{Q}(\theta)} = \mathbf{Z}[\alpha]$ *for an* $\alpha \in \mathbf{L}$*, then the factorization of the ideal* $2\mathscr{O}_{\mathbf{L}}$ *would corresponds to the factorization of the minimal polynomial* f_α *of* α *in* $\mathbf{F}_2[X]$*. But* 2 *splits in* $\mathscr{O}_{\mathbf{L}}$*: its prime factors are:* $(1/2(\theta^2 - \theta) + 1)\mathscr{O}_{\mathbf{L}}$*,* $(-\theta^2 + 2\theta - 3)\mathscr{O}_{\mathbf{L}}$ *and* $(-3/2\theta^2 + 6/2\theta - 4)\mathscr{O}_{\mathbf{L}}$ *implying that the factoring of* f_α *must have to 3 distinct linear irreducible factors. But there are only two irreducible linear polynomials in* $\mathbf{F}_2[X]$*, a contradiction.*

Remark. *Given a defining polynomial of a number field* $\mathbf{L}$*, the problem of determining the ring* $\mathscr{O}_{\mathbf{L}}$ *is computationally equivalent (up-to additional polynomial computations) to the problem of finding the largest square-free of the discriminant (see Section 1.3.3 for a definition of this notion) of* $\mathbf{L}$*. But being able to retrieve the square-free part does not seem to be an easy problem, as there is, up to our knowledge, no significant speedups on the naive approach consisting in factorizing the discriminant, leading to a subexponential algorithm. However, Buchmann and Lenstra showed in [27] that we can* approximate[2] *this ring of integers in polynomial time. This approximation can be used in place of* $\mathscr{O}_{\mathbf{L}}$ *for certain applications in algorithmic number theory.*

1.3 NORM, TRACE AND DISCRIMINANTS

1.3.1 Norm and trace

Definition 1.3.1 (Norm and trace)**.** *Let* $\mathbf{L}/\mathbf{K}$ *be a field extension, and* $\alpha \in \mathbf{L}$*. We write* $m_\alpha : \mathbf{L} \to \mathbf{L}$ *for the "multiplication by* α*" map:* $\ell \mapsto \alpha\ell$*. Viewing this as a linear map of* $\mathbf{K}$ *vector spaces, we define the* norm *of the element* α *to be*

$$N_{\mathbf{L}/\mathbf{K}}(\alpha) = \det m_\alpha,$$

and the trace *to be*

$$\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(\alpha) = \mathrm{tr}\, m_\alpha.$$

The following property is immediate by properties of the determinant and trace of linear maps:

Proposition 1.3.1. *For a field extension* $\mathbf{L}/\mathbf{K}$ *and* $a, b \in \mathbf{L}$*, we have:*

$$N_{\mathbf{L}/\mathbf{Q}}(ab) = N_{\mathbf{L}/\mathbf{K}}(a)N_{\mathbf{L}/\mathbf{K}}(b) \qquad \text{and} \qquad \mathrm{tr}(a+b) = \mathrm{tr}(a) + \mathrm{tr}(b).$$

We can alternatively define the norm and trace as follows:

2 More precisely, the algorithm of Buchmann and Lenstra takes as input an order A of $\mathscr{O}_{\mathbf{L}}$ and generates in polynomial time an order B containing A and a number q such that q is squarefree iff $B = \mathscr{O}_{\mathbf{L}}$. Equivalently given an order A and a squarefree integer m, this allows to devise in polynomial time an order B containing A such that $\gcd(m, [\mathscr{O}_{\mathbf{L}} : B]) = 1$.

Proposition 1.3.2. *Let $m_\alpha \in \mathbf{K}[X]$ be the minimal polynomial of α. Define its characteristic polynomial χ_α to be*

$$\det(XI - m_\alpha) = m_\alpha^{[\mathbf{L}:\mathbf{K}(\alpha)]}$$

Hence if $m_\alpha(x)$ splits in some field $\mathbf{L}' \supseteq \mathbf{K}(\alpha)$, as the product:

$$m_\alpha(x) = (x - \alpha_1) \cdots (x - \alpha_r),$$

then

$$N_{\mathbf{K}(\alpha)/\mathbf{K}}(\alpha) = \prod_{i=1}^{r} \alpha_i, \quad \mathrm{tr}_{\mathbf{K}(\alpha)/\mathbf{K}}(\alpha) = \sum_{i=1}^{r} \alpha_i,$$

and as such:

$$N_{\mathbf{L}/\mathbf{K}}(\alpha) = \left(\prod_{i=1}^{r} \alpha_i\right)^{[\mathbf{L}:\mathbf{K}(\alpha)]}, \quad \mathrm{tr}_{\mathbf{L}/\mathbf{K}} = [\mathbf{L} : \mathbf{K}(\alpha)]\left(\sum_{i=1}^{r} \alpha_i\right).$$

Remark. *This result implies that the trace and norm of an algebraic integer are* rational integers*, by Newton formulas on the relations between coefficients and roots.*

Example (Quadratic integers). *Let $\mathbf{L} = \mathbf{Q}\left(\sqrt{d}\right) \cong \mathbf{Q}[z]/(z^2 - d)$, where d is not a square in K. As a vector space over K, we can take $1, \sqrt{d}$ as our basis. So every α can be written as*

$$\alpha = x + y\sqrt{d}.$$

Hence the matrix of multiplication by α is

$$m_\alpha = \begin{pmatrix} x & d \cdot y \\ y & x \end{pmatrix}.$$

So the trace and norm are given by

$$\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(x + y\sqrt{d}) = 2x = (x + y\sqrt{d}) + (x - y\sqrt{d})$$
$$N_{\mathbf{L}/\mathbf{K}}(x + y\sqrt{d}) = x^2 - dy^2 = (x + y\sqrt{d})(x - y\sqrt{d})$$

We can also obtain directly this result by considering the roots of the minimal polynomial of $\alpha = x + y\sqrt{d}$, namely $(\alpha - x)^2 - y^2 d = 0$, which has roots $x \pm y\sqrt{d}$.

In particular, if $\mathbf{L} = \mathbf{Q}(\sqrt{d})$, with $d < 0$, then the norm of an element is just the square of its module viewed as an element of $\mathbf{C}$.

Remark. *When representing elements of $\mathbf{L} = \mathbf{Q}[X]/(P)$ by polynomials modulo (P) of degree n, a simple way to compute the norm of $\alpha \in \mathbf{L}$, is to compute the polynomial resultant of α and P, which can be done in*

$$O\big(n^2 \log N_{\mathbf{L}/\mathbf{K}}(\alpha) \log\log N_{\mathbf{L}/\mathbf{K}}(\alpha) + (\log N_{\mathbf{L}/\mathbf{K}}\alpha)^2\big)$$

operations.

1.3.2 Field embeddings

1.3.2.1. *Structure of the fields embeddings.* By being algebraically closed and containing $\mathbf{Q}$, every number field is a subfield of $\mathbf{C}$. A natural object of study is then the possible embeddings—that is field homomorphisms—$\mathbf{L} \hookrightarrow \mathbf{C}$.

Example. *For $\mathbf{Q}(\sqrt{-1})$, there are two such embeddings—one sends $\sqrt{-1}$ to i and the other sends $\sqrt{-1}$ to $-i$.*

This situation captures the general case: the rigidity imposed by the primitive element theorem ensures that all of the embeddings are completely determined by the image of a primitive element of $\mathbf{L}$.

Lemma 1.3.1. *The number of field embeddings $\mathbf{L} \hookrightarrow \mathbf{C}$ is equal to its degree $[\mathbf{L} : \mathbf{Q}]$.*

Proof. Let α be a primitive element, and $m_\alpha(x) \in \mathbf{Q}[X]$ be its minimal polynomial, so that $\deg m_\alpha = [\mathbf{L} : \mathbf{Q}] = n$. By primitivity, $(1, \alpha, \alpha^2, \dots, \alpha^{n-1})$ is a basis of $\mathbf{K}$ as $\mathbf{Q}$-vector space. Since $\mathbf{L}/\mathbf{Q}$ is separable as $\mathbf{Q}$ is of characteristic o, we know m_α has n distinct roots in $\mathbf{C}$, so that we write:

$$m_\alpha(x) = (x - \alpha_1) \cdots (x - \alpha_n).$$

Now an embedding $\mathbf{Q}[X]/(m_\alpha) \hookrightarrow \mathbf{C}$ is uniquely determined by the image of x, and x must be sent to one of the roots of m_α, since its image must satisfy

$$m(\sigma(\alpha)) = \sigma(m_\alpha(\alpha)) = 0.$$

So for each i, the map $x \mapsto \alpha_i$ gives us a field embedding and clearly none of them can be equal to another one, so that there are exactly n of them. ∎

We can separate the embeddings by theirs images:

Definition 1.3.2 (Signature of a number field)**.** *We write r for the number of real field embeddings $\mathbf{L} \hookrightarrow \mathbf{R}$, and s the number of pairs of non-real field embeddings $\mathbf{L} \hookrightarrow \mathbf{C}$. Then*

$$[\mathbf{L} : \mathbf{Q}] = r + 2s.$$

Alternatively, r is the number of real roots of m_α, and s is the number of pairs of complex conjugate roots.

1.3.2.2. *Connection to the norm and trace.* Using these field embeddings, we can give an alternative definition of the trace and norm forms, which reveals directly their respective and additive natures:

Lemma 1.3.2. *Let $\mathbf{L}/\mathbf{Q}$ be a number field. If $\sigma_1, \dots, \sigma_n : \mathbf{L} \to \mathbf{C}$ are the different field embeddings and $\beta \in \mathbf{L}$, then*

$$\mathrm{tr}_{\mathbf{L}/\mathbf{Q}}(\beta) = \sum_i \sigma_i(\beta), \quad N_{\mathbf{L}/\mathbf{Q}}(\beta) = \prod_i \sigma_i(\beta).$$

We call $\sigma_1(\beta), \dots, \sigma_n(\beta)$ the conjugates *of β in $\mathbf{C}$.*

1.3.2.3. *Characterization of units.* Using this definition of the norm, we can give an alternative characterization of the units of $\mathbf{L}$:

Lemma 1.3.3. *Let $x \in \mathscr{O}_{\mathbf{L}}$. Then x is a unit if and only if $|N_{\mathbf{L}/\mathbf{Q}}(x)| = 1$.*

Proof. Let $x \in \mathscr{O}_{\mathbf{L}}^{\times}$, then there is some $y \in \mathscr{O}_{\mathbf{L}}$ such that $xy = 1$. Taking the norm yields $N_{\mathbf{L}/\mathbf{Q}}(x)N_{\mathbf{L}/\mathbf{Q}}(y) = 1$. So $N_{\mathbf{L}/\mathbf{Q}}(x)$ is a unit in $\mathbf{Z}$, i.e. ± 1.

Reciprocally, let $\sigma_1, \dots, \sigma_n : \mathbf{L} \to \mathbf{C}$ be the n embeddings of $\mathbf{L}$ in $\mathbf{C}$. Without loss of generality, up to composing by σ_1^{-1}, we can identify $\mathbf{L}$ with a subfield of $\mathbf{C}$, so that σ_1 is the inclusion map. Then for each $x \in \mathscr{O}_{\mathbf{L}}$, we have

$$N_{\mathbf{L}/\mathbf{Q}}(x) = x\sigma_2(x) \cdots \sigma_n(x).$$

Now if $N_{\mathbf{L}/\mathbf{Q}}(x) = \pm 1$, then $x^{-1} = \pm\sigma_2(x) \cdots \sigma_n(x)$. So we have $x^{-1} \in \mathscr{O}_{\mathbf{L}}$, since this is a product of algebraic integers. So x is a unit in $\mathscr{O}_{\mathbf{L}}$. ∎

Corollary 1.3.1. *If $x \in \mathscr{O}_{\mathbf{L}}$ is such that $N_{\mathbf{L}/\mathbf{Q}}(x)$ is prime, then x is irreducible.*

Proof. If $x = ab$, then $N_{\mathbf{L}/\mathbf{Q}}(a)N_{\mathbf{L}/\mathbf{Q}}(b) = N_{\mathbf{L}/\mathbf{Q}}(x)$. Since $N_{\mathbf{L}/\mathbf{Q}}(x)$ is prime, either $N_{\mathbf{L}/\mathbf{Q}}(a) = \pm 1$ or $N_{\mathbf{L}/\mathbf{Q}}(b) = \pm 1$. So a or b is a unit. ∎

1.3.3 Discriminant

The final invariant we attach to a number field is its discriminant. In some sense it encodes the size of its ring of integers. It is based on the following observation:

Proposition 1.3.3. *Let $\mathbf{L}/\mathbf{Q}$ be a number field. Then the $\mathbf{L}$-bilinear form $\mathbf{L} \times \mathbf{L} \to \mathbf{Q}$ defined by $(x, y) \mapsto \mathrm{tr}_{\mathbf{L}/\mathbf{Q}}(xy)$ is non-degenerate. Equivalently, if $(\alpha_1, \dots, \alpha_n)$ is a basis of $\mathbf{L}$, then the Gram matrix*

$$\Delta(\alpha_1, \dots, \alpha_n) = \det\left(\mathrm{tr}_{\mathbf{L}/\mathbf{Q}}(\alpha_i\alpha_j)\right)_{1 \le i,j \le n}.$$

has non-zero determinant.

Proof. Let $\sigma_1, \dots, \sigma_n : \mathbf{L} \to \mathbf{C}$ be the n distinct $\mathbf{L}$-linear field embeddings to $\mathbf{C}$. Define the embedding matrix

$$S = (\sigma_i(\alpha_j))_{i,j=1,\dots,n} = \begin{pmatrix} \sigma_1(\alpha_1) & \cdots & \sigma_1(\alpha_n) \\ \vdots & \ddots & \vdots \\ \sigma_n(\alpha_1) & \cdots & \sigma_n(\alpha_n). \end{pmatrix},$$

so that a simple computation yields

$$\begin{aligned} S^T S &= \left(\sum_{k=1}^{n} \sigma_k(\alpha_i)\sigma_k(\alpha_j) \right)_{i,j} \\ &= \left(\sum_{k=1}^{n} \sigma_k(\alpha_i\alpha_j) \right)_{i,j} \\ &= \left(\mathrm{tr}_{\mathbf{L}/\mathbf{Q}}(\alpha_i\alpha_j)\right)_{i,j} \\ &= \Delta(\alpha_1, \dots, \alpha_n), \end{aligned}$$

since σ_k is a field homomorphism. Then $\det \Delta(\alpha_1, \dots, \alpha_n) = (\det S)^2$. By the primitive element theorem, write $\mathbf{L} = \mathbf{Q}(\theta)$ such that $1, \theta, \dots, \theta^{n-1}$ is a basis for $\mathbf{L}$ over $\mathbf{Q}$, with $[\mathbf{L} : \mathbf{Q}] = n$. Now S becomes

$$S = \begin{pmatrix} 1 & \sigma_1(\theta) & \cdots & \sigma_1(\theta)^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_n(\theta) & \cdots & \sigma_n(\theta)^{n-1} \end{pmatrix}.$$

This is a Vandermonde matrix, and so

$$\Delta(1, \theta, \dots, \theta^{n-1}) = (\det S)^2 = \prod_{i<j} (\sigma_i(\theta) - \sigma_j(\theta))^2.$$

Since the field extension is separable, and hence $\sigma_i \neq \sigma_j$ for all i, j, this implies $\sigma_i(\theta) \neq \sigma_j(\theta)$, since θ generates the field. So the product is non-zero and as such the determinant is non-zero. The base change formula ensures the result for any other basis. ∎

The non degeneracy of this form now allows to prove that the ring of integers of a number field have a $\mathbf{Z}$-basis.

Proof of Theorem 1.2.2. Let $\alpha_1, \dots, \alpha_n$ be any basis of $\mathbf{L}$ over $\mathbf{Q}$. Then there are some $n_i \in \mathbf{Z}$ such that $n_i \alpha_i \in \mathscr{O}_\mathbf{L}$ (for instance by taking n_i to be a common denominator of coefficients of the minimal polynomial of α_i). So without loss of generality, we can suppose that $\alpha_1, \dots, \alpha_n \in \mathscr{O}_\mathbf{L}$, and are a basis of $\mathbf{L}$ over $\mathbf{Q}$. Since α_i are integral, so are $\alpha_i \alpha_j$, and so all of these products have integer trace. Hence $\Delta(\alpha_1, \dots, \alpha_n)$, being the determinant of a matrix with integer entries, is an integer.

Now choose a $\mathbf{Q}$-basis $\alpha_1, \dots, \alpha_n \in \mathscr{O}_\mathbf{L}$ such that $\Delta(\alpha_1, \dots, \alpha_n) \in \mathbf{Z} \setminus \{0\}$ has *minimal absolute value*. We now show that these are an integral basis, i.e. a basis of $\mathscr{O}_\mathbf{L}$ as $\mathbf{Z}$-module.

Let $x \in \mathscr{O}_\mathbf{L}$, and write $x = \sum_{i=1}^n \lambda_i \alpha_i$ for some $\lambda_i \in \mathbf{Q}$. These λ_i are necessarily unique since $\alpha_1, \dots, \alpha_n$ is a basis.

Suppose some $\lambda_i \notin \mathbf{Z}$. without loss of generality, we can say that $\lambda_1 \notin \mathbf{Z}$. We write

$$\lambda_1 = n_1 + \varepsilon_1,$$

for $n_1 \in \mathbf{Z}$ and $0 < \varepsilon_1 < 1$. Then

$$\alpha_1' = x - n_1 \alpha_1 = \varepsilon_1 \alpha_1 + \lambda_2 \alpha_2 + \cdots + \lambda_n \alpha_n \in \mathscr{O}_\mathbf{L}.$$

So $\alpha_1', \alpha_2, \dots, \alpha_n$ is still a basis for $\mathbf{L}/\mathbf{Q}$, and are still in $\mathscr{O}_\mathbf{L}$. But then

$$\Delta(\alpha_1', \dots, \alpha_n) = \varepsilon_1^2 \cdot \Delta(\alpha_1, \dots, \alpha_n) < \Delta(\alpha_1, \dots, \alpha_n),$$

which contradicts the minimality assumption. So we must have $\lambda_i \in \mathbf{Z}$ for all $\mathbf{Z}$, ensuring that $\alpha_1, \dots, \alpha_n$ is s an integral basis for $\mathscr{O}_\mathbf{L}$. ∎

Now if $\alpha'_1, \dots, \alpha'_n$ is another integral basis of $\mathbf{L}$ over $\mathbf{Q}$, then there is some $U \in \mathrm{Gl}_n(\mathbf{Z})$ such that $g\alpha_i = \alpha'_i$. Since $\det(U)$ is invertible in $\mathbf{Z}$, it must be 1 or -1, and hence

$$\det \Delta(\alpha'_1, \dots, \alpha'_n) = \det(U)^2 \Delta(\alpha_1, \dots, \alpha_n) = \Delta(\alpha_1, \dots, \alpha_n)$$

and is independent of the choice of *integral* basis.

Definition 1.3.3 (Discriminant)**.** *The* discriminant $\Delta_{\mathbf{L}}$ *of a number field* $\mathbf{L}$ *is defined as*

$$\Delta_{\mathbf{L}} = \Delta(\alpha_1, \dots, \alpha_n)$$

for any integral basis $\alpha_1, \dots, \alpha_n$*.*

1.4 MULTIPLICATIVE STRUCTURE OF IDEALS

Again, let $\mathbf{L}/\mathbf{Q}$ be a number field. It turns out that in general, the integral ring $\mathscr{O}_{\mathbf{L}}$ is not as simple as the rational integers $\mathbf{Z}$. In particular, the unique factorization property fails in general.

Example. *Let* $\mathbf{L} = \mathbf{Q}(\sqrt{-5})$*. Then* $\mathscr{O}_{\mathbf{L}} = \mathbf{Z}[\sqrt{-5}]$*. Then we find by calculation:*

$$3 \cdot 7 = (1 + 2\sqrt{-5})(1 - 2\sqrt{-5}).$$

Note that it is still possible to factor any element into a product of irreducibles, this fact being proved trivially by induction on the norm. To fix the lack of unique factorization, we instead look at ideals in $\mathscr{O}_{\mathbf{L}}$. Recall from Paragraph 1.1.5.1 that ideals have a natural multiplicative structure. We now prove that the unique factorization property is actually verified *for ideals*, that is to say, in the light of Paragraph 1.1.5.4:

Theorem 1.4.1. *Let* $\mathbf{L}/\mathbf{Q}$ *be a number field, and* $\mathscr{O}_{\mathbf{L}}$ *be its ring of integers. Then* $\mathscr{O}_{\mathbf{L}}$ *is a Dedekind domain.*

Remark. *Kummer first published the failure of unique factorization in cyclotomic fields around 1844. While it is widely believed that Kummer was led to his* ideal complex numbers *by his interest in Fermat's Last Theorem, it occurs that he was actually more interested in higher reciprocity laws, which he considered to be* "the principal subject and the pinnacle of contemporary number theory"*. In modern language, a Kummer's "ideal number" is an algebraic integer which represents an ideal in the ring of integers of a number field (in the light of the now-called class field theory, every ideal is therefore represented by an ideal number in a certain field extension). The adjective* ideal *was coined to refer to the good arithmetical properties of these numbers, such as their unique factorization. These ideas have been refined to the general case by Kronecker and Dedekind, independently, during the next forty years, leading to major contributions, such as the modern notions of ideals, modules and divisors amongst others.*

To prove this maximality property of prime ideals, we need the following lemma:

Lemma 1.4.1. *Let $\mathfrak{a} \subseteq \mathscr{O}_\mathbf{L}$ be a non-zero ideal. Then $\mathfrak{a} \cap \mathbf{Z} \neq \{0\}$ and $\mathscr{O}_\mathbf{L}/\mathfrak{a}$ is finite.*

Proof. Let $\alpha \in \mathfrak{a}$ and $\alpha \neq 0$. Let $\chi_\alpha = x^m + a_{m-1}x^{m-1} + \cdots + a_0 \in \mathbf{Z}[X]$ be its minimal polynomial. We know $a_0 \neq 0$ as χ_α is irreducible. Since $\chi_\alpha(\alpha) = 0$, we have

$$a_0 = -\alpha(\alpha^{m-1} + a_{m-1}\alpha^{m-2} + \cdots + a_2\alpha + a_1).$$

We know that $\alpha \in \mathfrak{a} \subseteq \mathscr{O}_\mathbf{L}$ by assumption, and as such that the sum $\alpha^{m-1} + a_{m-1}\alpha^{m-2} + \cdots + a_2\alpha + a_1$ is in $\mathscr{O}_\mathbf{L}$. As such the whole right hand side is in $\mathfrak{a}$. But $a_0 \in \mathbf{Z}$ and therefore $a_0 \in \mathbf{Z} \cap \mathfrak{a}$.

Thus, $\langle a_0 \rangle \subseteq \mathfrak{a}$, inducing a surjection

$$\mathscr{O}_\mathbf{L}/\langle a_0 \rangle \longrightarrow \mathscr{O}_\mathbf{L}/\mathfrak{a}.$$

Hence, it suffices to show that $\mathscr{O}_\mathbf{L}/\langle a_0 \rangle$ is finite. Remark that for $d \in \mathbf{Z}$, we have that:

$$\mathscr{O}_\mathbf{L}/\langle d \rangle = \mathbf{Z}^n/d\mathbf{Z}^n = \left(\mathbf{Z}/d\mathbf{Z}\right)^n,$$

which is of course finite. ∎

Proof of Theorem 1.4.1. **Integrality:** Obvious, since $\mathscr{O}_\mathbf{L} \subseteq \mathbf{L}$.

Notherian: We showed that as an abelian group, $\mathscr{O}_\mathbf{L} \cong \mathbf{Z}^n$. So if $\mathfrak{a} \subseteq \mathscr{O}_\mathbf{L}$ is an ideal, then $\mathfrak{a} \subseteq \mathbf{Z}^n$ as a subgroup. So it is finitely generated as an abelian group, and hence finitely generated as an ideal.

Integrally closed: note that $\operatorname{Frac} \mathscr{O}_\mathbf{L} = \mathbf{L}$. If $x \in \mathbf{L}$ is integral over $\mathscr{O}_\mathbf{L}$, as $\mathscr{O}_\mathbf{L}$ is integral over $\mathbf{Z}$, x is also integral over $\mathbf{Z}$. So $x \in \mathscr{O}_\mathbf{L}$, by definition of $\mathscr{O}_\mathbf{L}$. ∎

Maximality of prime ideals: Let $\mathfrak{p}$ be a prime ideal. Then $\mathscr{O}_\mathbf{L}/\mathfrak{p}$ is an integral domain. Since Lemma 1.4.1 says $\mathscr{O}_\mathbf{L}/\mathfrak{p}$ is finite, we deduce that $\mathscr{O}_\mathbf{L}/\mathfrak{p}$ is a field[3]. So $\mathfrak{p}$ is maximal by definition.

1.5 NORMS OF IDEALS

1.5.1 Definition and multiplicativity

Definition 1.5.1 (Norm of ideals)**.** *Let $\mathfrak{a} \subseteq \mathscr{O}_\mathbf{L}$ be an ideal. We define*

$$|N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})| = \left|\mathscr{O}_\mathbf{L}/\mathfrak{a}\right| \in \mathbf{N}.$$

We already proved that $|\mathscr{O}_\mathbf{L}/\mathfrak{a}|$ is finite, so that this definition actually makes sense.

3 Classically we know that a finite integral domain must be a field. Indeed Let R be a finite integral domain and $x \in R$ with $x \neq 0$. Then the map $m_x : y \mapsto xy$ is injective, as having a trivial kernel since R is an integral domain. So it is a bijection by finiteness. And as such there exists some $y \in R$ such that $xy = 1$.

Remark. *It is clear that* $N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a}) = 1$ *if and only if* $\mathfrak{a} = \mathscr{O}_{\mathbf{L}}$ *(i.e.* $\mathfrak{a}$ *is a "unit").*

Example. *Let* $d \in \mathbf{Z}$*. Then since* $\mathscr{O}_{\mathbf{L}} \cong \mathbf{Z}^n$*, we have* $d\mathscr{O}_{\mathbf{L}} \cong (d\mathbf{Z})^n$*. So we have*

$$N_{\mathbf{L}/\mathbf{Q}}(\langle d \rangle) = |\mathbf{Z}^n/(d\mathbf{Z})^n| = |\mathbf{Z}/d\mathbf{Z}|^n = d^n.$$

Interestingly, the norm of any ideal is actually contained in the ideal itself:

Proposition 1.5.1. *For any ideal* $\mathfrak{a}$*, we have* $N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a}) \in \mathfrak{a} \cap \mathbf{Z}$*.*

Proof. It suffices to show that $N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a}) \in \mathfrak{a}$. Viewing $\mathscr{O}_{\mathbf{L}}/\mathfrak{a}$ as an additive group, the order of 1 is a factor of $N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})$. So $N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a}) = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a}) \cdot 1 = 0 \in \mathscr{O}_{\mathbf{L}}/\mathfrak{a}$. Hence $N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a}) \in \mathfrak{a}$. ∎

We now turn to the multiplicativity of the norm form:

Proposition 1.5.2. *Let* $\mathfrak{a}, \mathfrak{b} \subseteq \mathscr{O}_{\mathbf{L}}$ *be ideals. Then*

$$N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a}\mathfrak{b}) = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{b}).$$

Proof. By unique factorization theorem, it is enough to show that

$$N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{p}_1^{a_1} \cdots \mathfrak{p}_r^{a_r}) = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{p}_1)^{a_1} \cdots N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{p}_r)^{a_r}.$$

By the Chinese remainder theorem, we have

$$\mathscr{O}_{\mathbf{L}}\big/_{\mathfrak{p}_1^{a_1} \cdots \mathfrak{p}_r^{a_r}} \cong \mathscr{O}_{\mathbf{L}}\big/_{\mathfrak{p}_1^{a_1}} \times \cdots \times \mathscr{O}_{\mathbf{L}}\big/_{\mathfrak{p}_r^{a_r}}$$

where $\mathfrak{p}_1, \ldots, \mathfrak{p}_r$ are distinct prime ideals. But now we have:

$$\left|\mathscr{O}_{\mathbf{L}}\big/_{\mathfrak{p}^r}\right| = \left|\mathscr{O}_{\mathbf{L}}\big/_{\mathfrak{p}}\right| \times \left|\mathfrak{p}\big/_{\mathfrak{p}^2}\right| \times \cdots \times \left|\mathfrak{p}^{r-1}\big/_{\mathfrak{p}^r}\right| = \left|\mathscr{O}_{\mathbf{L}}\big/_{\mathfrak{p}}\right|^r,$$

since $\mathfrak{p}^k/\mathfrak{p}^{k+1}$ is a 1-dimensional vector space over the field $\mathscr{O}_{\mathbf{L}}/\mathfrak{p}$. ∎

1.5.2 Relation to the discriminant

We now show that the norm is actually related to the discriminant, introduced in Section 1.3.3. Recall that the discriminant is defined by:

$$\Delta(\alpha_1, \ldots, \alpha_n) = \det(\mathrm{tr}_{\mathbf{L}/\mathbf{Q}}(\alpha_i\alpha_j)) = \det(\sigma_i(\alpha_j))^2.$$

Proposition 1.5.3. *Let* $\mathfrak{a} \subseteq \mathscr{O}_{\mathbf{L}}$ *be an ideal,* $n = [\mathbf{L} : \mathbf{Q}]$*. Then*

1. *There exists* $\alpha_1, \ldots, \alpha_n \in \mathfrak{a}$ *such that*

$$\mathfrak{a} = \left\{ \sum_{i=1}^n r_i\alpha_i : r_i \in \mathbf{Z} \right\} = \bigoplus_{i=1}^n \alpha_i\mathbf{Z},$$

and $\alpha_1, \ldots, \alpha_n$ *are a basis of* $\mathbf{L}$ *over* $\mathbf{Q}$*.*

2. *For any such* $\alpha_1, \dots, \alpha_n$,

$$\Delta(\alpha_1, \dots, \alpha_n) = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})^2 \Delta_{\mathbf{L}}.$$

Remark. *The first point of Proposition 1.5.3 states that any ideal* $\mathfrak{a}$ *of* $\mathscr{O}_{\mathbf{L}}$ *has a natural structure of* free $\mathbf{Z}$*-module of rank* n*. Since the ring of integers is a Dedekind domain, we also know from Section 1.1 that it also has a structure of* projective $\mathscr{O}_{\mathbf{L}}$*-module since it is generated by two elements.*

To prove Proposition 1.5.3, we need the following classical lemma on abelian groups:

Lemma 1.5.1. *Let* $\mathcal{M}$ *be a* $\mathbf{Z}$*-module, and suppose* $\mathcal{M} \subseteq \mathbf{Z}^n$ *is a subgroup of* $\mathbf{Z}^n$*. Then* $\mathcal{M} \cong \mathbf{Z}^r$ *for some* $0 \leq r \leq n$*. Moreover, if* $r = n$*, then we can choose a basis* $v_1, \dots, v_n$ *of* M *such that the change of basis matrix* $A = (a_{ij}) \in \mathbf{Z}^{n \times n}$ *is upper triangular, where*

$$v_j = \sum_{i=1}^{n} a_{ij} e_i,$$

for $e_1, \dots, e_n$ *is the standard basis of* $\mathbf{Z}^n$ *and as such we have:*

$$\left| \mathbf{Z}^n / \mathcal{M} \right| = |a_{11} a_{22} \cdots a_{nn}| = |\det A|.$$

Proof of Proposition 1.5.3. Let $d \in \mathfrak{a} \cap \mathbf{Z}$, where $d = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})$. Then $d\mathscr{O}_{\mathbf{L}} \subseteq \mathfrak{a} \subseteq \mathscr{O}_{\mathbf{L}}$. As abelian groups, after picking an integral basis $\alpha_1', \dots, \alpha_n'$ of $\mathscr{O}_{\mathbf{L}}$, we have

$$\mathbf{Z}^n \cong d\mathbf{Z}^n \subseteq \mathfrak{a} \subseteq \mathbf{Z}^n.$$

So $\mathfrak{a} \cong \mathbf{Z}^n$. Then the lemma gives us a basis $\alpha_1, \dots, \alpha_n$ of $\mathfrak{a}$ as a $\mathbf{Z}$-module. As a $\mathbf{Q}$-module, since the α_i are obtained from linear combinations of α_i', by basic linear algebra, $\alpha_1, \cdots, \alpha_n$ is also a basis of $\mathbf{L}$ over $\mathbf{Q}$.

Moreover, we know that we have

$$\Delta(\alpha_1, \dots, \alpha_n) = \det(A)^2 \Delta(\alpha_1', \cdots, \alpha_n').$$

Since $\det(A)^2 = \left| \mathscr{O}_{\mathbf{L}} / \mathfrak{a} \right|^2 = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})$ and $\Delta_{\mathbf{L}} = \Delta(\alpha_1', \dots, \alpha_n')$ by definition, the second part follows. ∎

At this point we have two different norms on elements. Given $\alpha \in \mathscr{O}_{\mathbf{L}}$, we can take the norm of the principal ideal generated by the element α: $N_{\mathbf{L}/\mathbf{Q}}(\langle \alpha \rangle)$, or consider the algebraic $N_{\mathbf{L}/\mathbf{Q}}(\alpha)$. They are actually equal, up to sign:

Lemma 1.5.2. *If* $\alpha \in \mathscr{O}_{\mathbf{L}}$*, then*

$$N_{\mathbf{L}/\mathbf{Q}}(\langle \alpha \rangle) = |N_{\mathbf{L}/\mathbf{Q}}(\alpha)|.$$

Proof. Let $\alpha_1, \dots, \alpha_n$ be an integral basis of $\mathscr{O}_{\mathbf{L}}$. Then $\alpha \cdot \alpha_1, \dots, \alpha \cdot \alpha_n$ is obviously an integral basis of $\langle \alpha \rangle$. So by Proposition 1.5.3 we find:

$$\Delta(\alpha\alpha_1, \dots, \alpha\alpha_n) = N_{\mathbf{L}/\mathbf{Q}}(\langle \alpha \rangle)^2 \Delta_{\mathbf{L}}.$$

But by definition of Δ we have:

$$\begin{aligned}
\Delta(\alpha\alpha_1, \dots, \alpha\alpha_n) &= \det\big((\sigma_i(\alpha\alpha_j))_{1\le i,j\le n}\big)^2 \\
&= \det\big((\sigma_i(\alpha)\sigma_i(\alpha_j))_{1\le i,j\le n}\big)^2 \\
&= \left(\prod_{i=1}^{n} \sigma_i(\alpha)\right)^2 \Delta(\alpha_1, \cdots, \alpha_n) \\
&= N_{\mathbf{L}/\mathbf{Q}}(\alpha)^2 \Delta_{\mathbf{L}}.
\end{aligned}$$

So

$$N_{\mathbf{L}/\mathbf{Q}}(\alpha)^2 = N_{\mathbf{L}/\mathbf{Q}}(\langle \alpha \rangle)^2.$$

But $N_{\mathbf{L}/\mathbf{Q}}(\langle \alpha \rangle)$ is positive. So the result follows. ∎

1.5.3 Computation with ideals

Since by the results of Section 1.1 an ideal $\mathfrak{a}$ is a projective module of rank 2 over $\mathscr{O}_{\mathbf{L}}$, we can represent it by a projective basis, that is a pair of elements in $\mathscr{O}_{\mathbf{L}}$ generating $\mathfrak{a}$. This is the *two-elements* representation. As it is also a free $\mathbf{Z}$-module of rank $[\mathbf{L} : \mathbf{Q}]$ we can represent $\mathfrak{a}$ by a $\mathbf{Z}-$generating family in a given basis of $\mathscr{O}_{\mathbf{L}}$, that is by a matrix with integral coefficients. This is the matrix representation of $\mathfrak{a}$.

1.5.3.1. *Changing the representation.* Given one of the two representations evoked above, we can switch to the other. Let $\mathfrak{a}$ an ideal of the number field $\mathbf{L}$. Suppose that $\mathscr{O}_{\mathbf{L}}$ is given by the $\mathbf{Z}$-basis $\omega_1, \dots, \omega_n$. Then if $\mathfrak{a}$ is generated by elements a and b, it is also spanned by $a\omega_1, \dots, a\omega_n, b\omega_1, \dots, a\omega_n$ as a $\mathbf{Z}$-module. It then suffices to compute the HNF[4] of the corresponding matrix to recover the coefficients of a basis of $\mathfrak{a}$ in $\omega_1, \dots, \omega_n$. Getting a two-element representation is more complicated, as we need to find uniformizers of primes, as implicitly shown in the proof of Proposition 1.1.3. A simple randomized algorithm is nonetheless practical: given a a generator of $\mathfrak{a} \cap \mathbf{Z}$, we sample uniformly at random $b \in a\mathscr{O}_{\mathbf{L}}$ and check if b suits, by checking of the HNF of $(b\mathscr{O}_{\mathbf{L}} \mid a\mathrm{Id})$ is equal to the HNF of $\mathfrak{a}$. The success probability of this procedure is at least $\prod_{\mathfrak{p}|a}\left(1 - \frac{1}{N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{p})}\right)$. This probability can be made independent of a with the techniques of Belabas [12].

4 The precise introduction of the Hermite Normal Form (HNF) is done in Section 2.2.1. In short, there exists a polynomial time algorithm allowing to compute this form, which in particular allows to reduce find a basis of a free $\mathbf{Z}$-module from an arbitrary generating family.

1.5.3.2. *Ideal arithmetic with matrices.* Given $\mathfrak{a}$, $\mathfrak{b}$ two ideals, represented respectively as matrices A and B of integers, the definition of sum and multiplication of ideals yield a simple expression for the generating families of these ideals. Indeed, it is clear that the families of vectors

$$\{A_1, \ldots, A_n, B_1, \ldots, B_n\}, \{A_i \cdot B_j \mid 1 \leq i, j \leq n\}$$

generates $\mathfrak{a} + \mathfrak{b}$ and $\mathfrak{ab}$. Then a HNF reduction reduces such families to basis. This implies that performing an addition of ideals can be done by computing a Hermite normal form from a matrix of size $n \times 2n$ and the multiplication from a computation of HNF of size $n \times n^2$.

The inversion of $\mathfrak{a}$ boils down to the computation of the *different ideal*, that is the dual $\mathbf{Z}$-module of $\mathscr{O}_{\mathbf{L}}$:

$$\mathfrak{d}(\mathbf{L}) = \{x \in \mathbf{L},\ \mathrm{tr}_{\mathbf{L}/\mathbf{Q}}(x\mathscr{O}_{\mathbf{L}}) \subset \mathbf{Z}\}.$$

Indeed, remark that a $\mathbf{Z}-$basis of $\mathfrak{a}^{-1}\mathfrak{d}(\mathbf{L})^{-1}$ is given by $(A^T T)^{-1}$, where T is the matrix $\left(\mathrm{tr}(\omega_i\omega_j)\right)_{i,j}$. Therefore to compute a basis of $\mathfrak{a}^{-1}$, we start by computing T^{-1}, a basis of the inverse of the different, then we compute a basis, denoted by N of $\mathfrak{a}\mathfrak{d}(\mathbf{L})^{-1}$ using the above-explained HNF method from A and T^{-1}. And so the columns of $(N^T T)^{-1}$ are a basis of $(\mathfrak{a}\mathfrak{d}(\mathbf{L})^{-1})^{-1}\mathfrak{d}(\mathbf{L})^{-1} = \mathfrak{a}^{-1}$. The cost of the inversion is then a constant number of multiplications of ideals.

1.5.3.3. *Going further with two-elements representation.* It would be interesting to be able to use the same approach for the two elements representation, since the set of sums, resp. of products, of the generators is only 4, instead of n^2. But then, we need to be able to reduce efficiently a generating family into a basis of the corresponding ideal. Using, for instance, the generalized HNF algorithm over number fields of [15] would actually be slower than the previous approach. We address this particular question through the prism of lattice reduction in Chapter 5.

The representation with two elements allows an easy inversion, as we can exploit the relation:

$$\mathfrak{a} \cap \mathfrak{b} = \mathfrak{ab}(\mathfrak{a} + \mathfrak{b})^{-1},$$

valid for any ideals $\mathfrak{a}, \mathfrak{b}$ of the number field $\mathbf{L}$. Indeed suppose that we want to invert an ideal $\mathfrak{c}$ generated by $\alpha, \beta \in \mathbf{L}$, then we have:

$$(\alpha\mathscr{O}_{\mathbf{L}} \cap \beta\mathscr{O}_{\mathbf{L}}) = \alpha\beta\mathscr{O}_{\mathbf{L}}(\alpha\mathscr{O}_{\mathbf{L}} + \beta\mathscr{O}_{\mathbf{L}})^{-1} = \alpha\beta\mathfrak{c}^{-1},$$

so that $\mathfrak{c}^{-1}$ is the intersection of the ideals generated by α^{-1} and β^{-1}

1.6 CYCLOTOMIC EXTENSIONS

In this final section we recall basic facts on cyclotomic fields. These particular number fields are very interesting since they are well understood and allow very explicit computations of invariants. For instance, the discriminant of a cyclotomic field has a very simple shape and its ring of integers

is very convenient to describe. For a very complete account on this subject, we invite the reader to refer to the monograph of Washington [163].

Definition 1.6.1 (Cyclotomic extension)**.** *The n-th* cyclotomic extension *of $\mathbf{Q}$, is the splitting field of the polynomial $X^n - 1$. We call n the* conductor *of the extension.*

Let $n \geq 2$ and let $\mathbf{L}$ be the n-th cyclotomic extension of $\mathbf{Q}$. The set of roots $\mathrm{Root}_{X^n-1}(\mathbf{L})$ is a subgroup of multiplicative group $\mathbf{L}^* = \mathbf{L} \setminus \{0\}$. Since this is a finite subgroup of $\mathbf{L}^*$, it is cyclic. But remark that the derivative $(X^n - 1)' = nX^{n-1}$ and this polynomial *can not* have a common root with $X^n - 1$. So $X^n - 1$ has no repeated roots: it has n *distinct roots.* So as a group,

$$\mathrm{Root}_{X^n-1}(\mathbf{L}) \cong \mathbf{Z}/n\mathbf{Z}.$$

In particular, this group has at least one element ζ_n of order n.

Definition 1.6.2 (Primitive root of unity)**.** *The n-th* primitive root of unity *is an element of order n in* $\mathrm{Root}_{X^n-1}(\mathbf{L})$.

These elements correspond to the elements of the multiplicative group of units in $\mathbf{Z}/n\mathbf{Z}$, written $\left(\mathbf{Z}/n\mathbf{Z}\right)^\times$.

Thus each root of unity can be reached from the primitive roots, meaning that we can define the cyclotomic extension with a polynomial of smaller degree, namely:

Theorem 1.6.1. *For each $d \in \mathbf{N}$, there exists a polynomial $\phi_d \in \mathbf{Z}[X]$ satisfying:*

1. *For each $n \in \mathbf{N}$, we have*

$$X^n - 1 = \prod_{d|n} \phi_d.$$

2. $\mathrm{Root}_{\phi_d}(\mathbf{L}) = \{d - th \text{ primitive roots of unity}\}$.

It is called the d-th cyclotomic polynomial.

Proof. Let us construct the cyclotomic polynomials by induction. When $n = 1$, let $\phi_1 = X - 1$. Then (i) and (ii) hold in this case, trivially. Assume now that (i) and (ii) hold for every $n' < n$. To fulfill (i) let us define:

$$f = \prod_{d|n, d<n} \phi_d.$$

A finite recurrence on this product with the induction hypothesis ensures that, $f \in \mathbf{Z}[X]$. Moreover, if $d \mid n$ and $d < n$, then $\phi_d \mid (X^n - 1)$ because $(X^d - 1) \mid (X^n - 1)$. To eventually prove that $f \mid X^n - 1$, it suffices to show that ϕ_d and $\phi_{d'}$ have no common roots for distinct $d, d' \mid n$ (and $d' < n$).

Indeed, ϕ_d and ϕ'_d have no common roots because by hypothesis:

$$\mathrm{Root}_{\phi_d}(\mathbf{L}) = \{d\text{th primitive roots of unity}\},$$
$$\mathrm{Root}_{\phi_{d'}}(\mathbf{L}) = \{d'\text{th primitive roots of unity}\},$$

and these two sets are disjoint (or else the roots would not be *primitive* by definition). Therefore ϕ_d and $\phi_{d'}$ do not have a common irreducible factor. Hence $f \mid X^n - 1$. So we can write

$$X^n - 1 = f\phi_n,$$

where $\phi_n \in \mathbf{Q}[X]$. Since f is monic, ϕ_n has integer coefficients. And as such $\phi_n \in \mathbf{Z}[X]$.

To prove the second part, note that by induction hypothesis,

$$\mathrm{Root}_f(\mathbf{L}) = \{\text{non-primitive } n\text{th roots of unit}\},$$

since all nth roots of unity are dth primitive roots of unity for some smaller d.

Since $f\phi_n = X^n - 1$, ϕ_n contains the remaining, primitive nth roots of unit. Since $X^n - 1$ has no repeated roots, we know that ϕ_n does not contain any extra roots. So

$$\mathrm{Root}_{\phi_n}(\mathbf{L}) = \{n\text{th primitive roots of unity}\}. \qquad \blacksquare$$

By counting the primitive roots, we have directly that for every $n \in \mathbf{N}$, $\deg \phi_n = \phi(n)$, where $\phi(n)$ is Euler totient function. Therefore we have:

Proposition 1.6.1. *Let $\mathbf{Q}[\zeta_n]$ the n-th cyclotomic extension, then $[\mathbf{Q}[\zeta_n] : \mathbf{Q}] = \varphi(n)$.*

The ring of integers of cyclotomic fields is very easy to describe, as the power base of the field $1, \zeta_n, \dots, \zeta_n^{n-1}$ is actually an *integral basis*:

Proposition 1.6.2. *Let $\mathbf{Q}[\zeta_n]$ the n-th cyclotomic extension, then*

$$\mathscr{O}_{\mathbf{Q}[\zeta_n]} = \mathbf{Z}[\zeta_n].$$

This allows to compute the discriminant of cyclotomic fields:

Theorem 1.6.2. *Let n be an non negative integer, then: The discriminant of $\mathbf{Q}[\zeta_n]$ is given by*

$$\Delta_n = (-1)^{\varphi(n)/2} \frac{n^{\varphi(n)}}{\prod_{p \mid n} p^{\varphi(n)/(p-1)}}$$

As this computation is quite lengthy and not enlightening for the rest of this manuscript, we let the reader refer to the monograph of Washington [163] for a complete proof of Proposition 1.6.2 and Theorem 1.6.2.

1.6.1 On the maximal real subfield of a cyclotomic

Given a number field $\mathbf{L}$, we can consider the set of *totally real subfields*, that is subfields with only real Archimedean embeddings. Since the compositum of totally real fields is also totally real, there exists a unique maximal element for the inclusion called the *maximal real subfield*.

Let us suppose that $\mathbf{L} = \mathbf{Q}[\zeta]$ is a cyclotomic field. We now prove that the maximal real subfield of $\mathbf{L}$ is easy to describe:

Lemma 1.6.1. *The subfield* $\mathbf{L}^+ = \mathbf{Q}\left(\zeta + \zeta^{-1}\right)$ *is the maximal real subfield of* $\mathbf{L} = \mathbf{Q}[\zeta]$.

Proof. Denote by f the conductor of $\mathbf{L}$. Then remark that

$$\zeta + \zeta^{-1} = 2\cos\left(\frac{2\pi}{f}\right) \in \mathbf{R},$$

so that $\mathbf{L}^+$ is totally real. Hence, $[\mathbf{L}^+ : \mathbf{L}] > 1$. Now set $f(X) = X^2 - (\zeta + \zeta^{-1})X + 1 = (X - \zeta)(X - \zeta^{-}1)$, so that f vanishes at ζ. Thus $[\mathbf{L}^+ : \mathbf{L}] \leq 2$, and we can conclude that $[\mathbf{L}^+ : \mathbf{L}] = 2$, concluding the proof. ∎

We have seen that the ring of integers of the cyclotomic field of conductor f is easily described as being $\mathbf{Z}[\zeta_f]$. The same description works for its maximal real subfield.

Proposition 1.6.3. *The ring of integer* $\mathscr{O}_{\mathbf{L}^+}$ *of the maximal real subfield* $\mathbf{L}^+ = \mathbf{Q}\left(\zeta + \zeta^{-1}\right)$ *of the cyclotomic field* $\mathbf{L} = \mathbf{Q}[\zeta]$ *satifsies:*

$$\mathscr{O}_{\mathbf{L}^+} = \mathbf{Z}[\zeta^{-1} + \zeta].$$

Proof. The inclusion $\mathbf{Z}[\zeta^{-1} + \zeta] \subseteq \mathscr{O}_{\mathbf{L}^+}$ is clear. Conversely, let us take $u \in \mathscr{O}_{\mathbf{L}^+} = \mathscr{O}_{\mathbf{L}} \cap \mathbf{L}^+$. Then we can decompose u in the power $\mathbf{Z}$-basis of $\mathscr{O}_{\mathbf{L}}$ and in the $\mathbf{Q}$-basis of the $(\zeta^i + \zeta^{-i})_i$:

$$u = \sum_{i=0}^{n-1} x_i \zeta^i$$
$$u = \sum_{i=0}^{(n-2)/2} y_i (\zeta^i + \zeta^{-i}),$$

so that $0 = \sum_{i=0}^{(n-2)/2} (x_i - y_i)\zeta^i + \sum_{i=(n-2)/2+1}^{n-1} (x_i - y_{n-1-i})\zeta^i$. We use the $\mathbf{Q}$-freeness of the power-basis to conclude that the y_i are all integers. ∎

2

GEOMETRY OF NUMBERS

This chapter presents the basic notions of classical and algorithmic geometry of numbers used in the subsequent parts of this manuscript. After recalling the definition of lattices and related geometric invariants, we move on to the reduction theory of Euclidean lattices with an emphasis on the algorithmic side of this theory, namely by studying the LLL algorithm and its generalization such as BKZ and DBKZ.

2.1 LATTICES

2.1.1 Euclidean spaces

2.1.1.1. *Bilinear forms.* In all of the following, we consider a finite-dimensional real vector space V. Its dual space, that is the vector space of linear forms over V, is denoted by $V^\vee$.

Definition 2.1.1 (Bilinear form)**.** *A (real) bilinear form over V is an application $b : V \times V \to \mathbf{R}$ so that:*

1. *For all $x \in V$, $b(x, \cdot) : y \mapsto b(x, y)$ is a linear map.*
2. *For all $y \in V$, $b(\cdot, y) : x \mapsto b(x, y)$ is a linear map.*

A bilinear form b over V is said to be:

Symmetric: when $\forall x, y \in V^2, b(y, x) = b(x, y)$,

Positive: when $\forall x \in V, b(x, x) \geq 0$,

Definite: when $\forall x \in V, b(x, x) = 0 \Leftrightarrow b = 0$.

A form satisfying these last three conditions is called an *inner product* over V.

Example.
- *On $\mathbf{R}^d$, the form $b[(x_1, \ldots, x_d), (y_1, \ldots, y_d)] = \sum_{i=1}^d x_i y_i$ is an inner product, called the* canonical *inner product of $\mathbf{R}^d$.*
- *Let $f, g \in V^\vee$ two linear forms, the product form $b : (x, y) \mapsto f(x)g(y)$ is clearly bilinear and symmetric, but not definite.*

2.1.2 Orthogonality relations

In this section V is a fixed real vector space of dimension d, and b a quadratic form acting on it.

Definition 2.1.2. *Two vectors $x, y \in V$ are called b-orthogonal (or simply orthogonal when there is no ambiguity on the form b considered) when $b(x,y) = 0$.*

This definition extends naturally to any couple of subsets of V: $A, B \subset V$ are orthogonal when any pair of vectors $(x,y) \in A \times B$ is orthogonal.

Definition 2.1.3. *Let $A \subset V$. The (b-)orthogonal of A is the subspace of V:*

$$A^{\perp} = \left\{ x \in V \;\middle|\; \forall a \in A, b(x,a) = 0 \right\}.$$

It is the largest (for the inclusion relation) space of V which is orthogonal to A.

2.1.2.1. *Euclidean space and its inner product.*

Definition 2.1.4 (Euclidean space)**.** *An Euclidean space is a pair $(V, \|\cdot\|)$ consisting of a finite dimensional real vector space V and an Euclidean norm $\|\cdot\|\colon V \to \mathbf{R}^+$, that is a norm[1] satisfying the parallelogram identity: $\|v+w\|^2 = 2\|v\|^2 + 2\|w\|^2 - \|v-w\|^2$ for every $v, w \in V$.*

Given a Euclidean space $(V, \|\cdot\|)$, we denote by

$$\langle \cdot\,, \cdot \rangle : \left| \begin{array}{ccc} V \times V & \longrightarrow & \mathbf{R} \\ x, y & \longmapsto & \frac{\|x+y\|^2 - \|x-y\|^2}{2} \end{array} \right. ,$$

its corresponding bilinear symmetric form. It is clear that this form is an inner product.

2.1.2.2. *Quotient of Euclidean spaces.* Let $(V, \|\cdot\|)$ be an Euclidean space, and let V' be a subspace of V. Then the restriction of $\|\cdot\|$ to V' endows V' with an Euclidean structure. The quotient space V/V' can be endowed with a canonical Euclidean structure by defining:

$$\|v + V'\| = \inf_{v' \in V'} \|v - v'\|,$$

for all $v \in V$. Geometrically, the vector v' minimizing the norm $\|v - v'\|$ is the orthogonal projection of v onto V'. Indeed, denoting by p the latter projection, we have for any $v' \in V'$:

$$\|v - v'\|^2 = \|v - p + (p - v')\|^2 = \|v - p\|^2 + \|p - v'\|^2 + 2\langle v - p, p - v' \rangle.$$

1 Recall that a norm $\|\cdot\|$ is a mapping from V to $[0, +\infty[$, which satisfies the three following properties:

Absolutly scalability: $\forall x \in V, \forall \alpha \in \mathbf{R}, \|\alpha v\| = |\alpha| \|v\|$,

Separation: $\forall x \in V \setminus \{0\}, \|x\| > 0$,

Triangular inequality: $\forall x, y \in V, \|x + y\| \leq \|x\| + \|y\|$.

But $v - p \in V'^{\perp}$ by definition of p, so that $\langle v - p, p - v' \rangle = 0$. Hence:

$$\|v - v'\|^2 = \|v - p\|^2 + \|p - v'\|^2 \geq \|v - p\|^2.$$

This quotient norm is well-defined and Euclidean since we can check that it derives from the restriction of the inner product of V on $V'^{\perp}$, as:

$$\langle v + V', w + V' \rangle = \langle \pi_{V'^{\perp}}(v), \pi_{V'^{\perp}}(w) \rangle.$$

2.1.3 Lattices

Definition 2.1.5 (Lattice). *A (real)* lattice Λ *is a finitely generated free* $\mathbf{Z}$*-module, endowed with a Euclidean norm on* $\|.\|$ *on the real vector space* $\Lambda_{\mathbf{R}} = \Lambda \otimes_{\mathbf{Z}} \mathbf{R}$.

From now on, $\Lambda \otimes_{\mathbf{Z}} 1$ is simply identified with Λ. By definition of a finitely-generated free module, there exists a finite family $(v_1, \ldots, v_d) \in \Lambda^d$ such that $\Lambda = \bigoplus_{i=1}^{d} v_i \mathbf{Z}$, called a *basis* of Λ. Every basis has the same number of elements $\operatorname{rk} \Lambda$, called the *rank of the lattice.*

Lemma 2.1.1. *The lattice* Λ *is discrete for the topology induced by* $\|\cdot\|_\Lambda$ *in the space* $\Lambda_{\mathbf{R}}$.

Proof. Let $(\Lambda, \|\cdot\|_\Lambda)$ be a lattice of rank d; its underlying $\mathbf{Z}$-module Λ being free, we have $\Lambda \cong \mathbf{Z}^d$, so that:

$$\Lambda_{\mathbf{R}} \cong \mathbf{Z}^d \otimes_{\mathbf{Z}} \mathbf{R} \cong \left(\oplus_{i=1}^{d} \mathbf{Z} \right) \otimes_{\mathbf{Z}} \mathbf{R} \cong \oplus_{i=1}^{d} (\mathbf{Z} \otimes_{\mathbf{Z}} \mathbf{R}) \cong \mathbf{R}^d.$$

Fix now a basis $(v_i)_{1 \leq i \leq d}$ of Λ; the family $(v_i \otimes 1)_{1 \leq i \leq d}$ is then a spanning set of cardinality d in a space of dimension d: it is a basis of $\Lambda_{\mathbf{R}}$. By the group structure of Λ it suffices to prove that the vector $0_\Lambda \otimes 1$ is an isolated point of $\Lambda \otimes 1$ to conclude. Let then $(x_k)_{k \in \mathbf{N}}$ be any sequence of $\Lambda \otimes 1$ converging towards $0_\Lambda \otimes 1$. Then, for each $k \in \mathbf{N}$, we have by decomposing:

$$x_k = \sum_{i=1}^{d} x_k^{(i)} (v_i \otimes 1),$$

which makes arise d sequences $\left(x_k^{(i)} \right)_{k \in \mathbf{N}}$ of integers. Since $(v_i \otimes 1)_{1 \leq i \leq d}$ is a basis, by characterization of the convergence in normed spaces, each of them converges towards o and is therefore stationary. As a consequence, $(x_k)_{k \in \mathbf{N}}$ is also stationary, entailing that $0_\Lambda \otimes 1$ is isolated. ∎

Conversely, it appears that this discreteness property characterize a lattice: any discrete additive subgroup of an Euclidean vector space is also a lattice, in the sense of definition Definition 2.1.5.

Lemma 2.1.2. *Let* V *an Euclidean vector space, and* Λ *be a discrete subgroup of* V*. Then,* Λ *is a free* $\mathbf{Z}$*-module of finite rank.*

Proof. Since we can replace V by the vector space spanned by Λ, we can suppose without loss of generality that Λ spans V itself. Let $v_1, \dots, v_k$ be a family of vectors of Λ which is $\mathbf{R}$-free, of maximal cardinality. Hence, $k = \dim(V)$ by our assumption. Then, $\Lambda' = v_1\mathbf{Z} \oplus \cdots \oplus v_k\mathbf{Z}$ is a subgroup of Λ by freeness hypothesis. Let us take any element $x \in \Lambda$, then, since we can identify the quotient $\Lambda \otimes \mathbf{R} / \Lambda' \otimes \mathbf{R}$ with the parallelotope $\mathcal{P}$ spanned by $v_1, \dots v_n$, we can write x as $\bar{x} + x'_\Lambda$ for $\bar{x} \in \mathcal{P}$ and $x'_\Lambda \in \Lambda'$. Thus $\bar{x} \in \Lambda$. But as Λ is discrete, so is the bounded set $\Lambda \cap \mathcal{P}$, which is therefore finite. As such, the quotient Λ / Λ' is finite, meaning that the index $\ell = [\Lambda : \Lambda']$ is also finite. Consequently, Λ is contained in $\frac{1}{\ell}\Lambda'$ and then is also a free $\mathbf{Z}$-module of rank n itself. ∎

Lemma 2.1.1 and Lemma 2.1.2 actually prove that an alternate definition of a lattice can be given, when know beforehand the ambient space:

Definition 2.1.6 (Lattice). *Let $(V, \|\cdot\|)$ be an Euclidean vector space. A (real)* lattice Λ *is a subgroup of $(V, +)$ which is discrete for the topology induced by $\|\cdot\|$.*

We may omit to write down the norm to refer to a lattice Λ when any ambiguity is removed by the context. However, as the following example shows, it might be crucial to keep track of it:

Example. *Let us consider the ring of integer of the number field $\mathbf{Q}[\sqrt{2}]$, which is $\mathbf{Z}[\sqrt{2}]$. A simple lattice structure can be given on this rank 2 module by setting $\|1\|_\alpha^2 = 1$, $\|\sqrt{2}\|_\alpha^2 = 2$ and $\langle 1, \sqrt{2}\rangle = \alpha$ for $\alpha \neq \sqrt{2}$, making this module discrete in $\mathbf{Z}[\sqrt{2}] \otimes_{\mathbf{Z}} \mathbf{R} \cong \mathbf{R}^2$. However, we shall point out that we can embed naturally $\mathbf{Z}[\sqrt{2}]$ into $\mathbf{R}$ as $\sqrt{2} \in \mathbf{R}$. The induced topology on $\mathbf{Z}[\sqrt{2}]$ makes it dense in $\mathbf{R}$, as $\sqrt{2}$ is an irrational number. We can not then refer to $\mathbf{Z}[\sqrt{2}]$ as a lattice, without specifying the corresponding inner product.*

2.1.3.1. *Sublattices, quotient lattice.* Let $(\Lambda, \|\cdot\|)$ be a lattice, and let Λ' be a submodule of Λ. Then the restriction of $\|\cdot\|$ to Λ' endows Λ with a lattice structure. The pair $(\Lambda', \|\cdot\|)$ is called a *sublattice* of Λ. Λ' is a pure sublattice if the quotient Λ / Λ' is torsion-free. In this case, it can be endowed with a canonical lattice structure by seeing it embedded in the quotient space $\Lambda_{\mathbf{R}} / \Lambda'_{\mathbf{R}}$, that is:

$$\|v + \Lambda'\| = \inf_{v' \in \Lambda'_{\mathbf{R}}} \|v - v'\|.$$

This definition makes sense since as Λ / Λ' is torsion-free over $\mathbf{Z}$, it is free.

Figure 1 presents an example of this situation for a planar lattice.

2.1.3.2. *Direct sums, tensor, and exterior products.* Let $(\Lambda, \|\cdot\|_\Lambda)$, $(\Lambda', \|\cdot\|_{\Lambda'})$ two lattices of rank-d and d'. The direct sum $\Lambda \oplus \Lambda'$ can be equipped with a lattice structure by setting: $\|v + v'\|^2 = \|v\|_\Lambda^2 + \|v'\|_{\Lambda'}^2$. The tensor product is also compatible with the lattice structure $\Lambda \otimes \Lambda'$ when setting:

$$\langle v \otimes v', w \otimes w' \rangle_{V \otimes V'} = \langle v, w \rangle_V \cdot \langle v', w' \rangle_{V'},$$

on pure tensors and extending it by bilinearity.

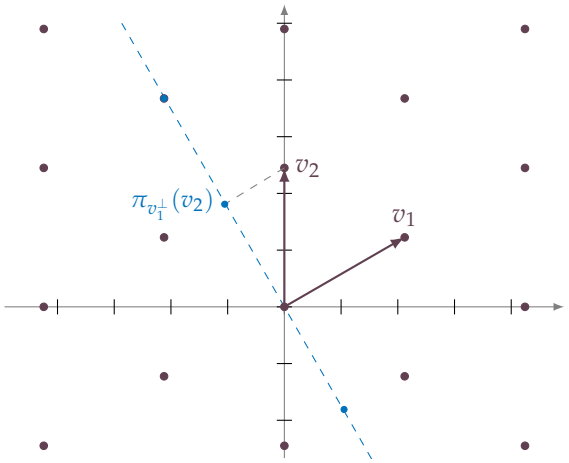


Figure 1: Two dimensional lattice Λ generated by $v_1, v_2 \in \mathbf{R}^2$, in purple, and the quotient lattice identified with $\pi_{v_1^\perp}(\Lambda)$ in blue. It is generated by $\pi_{v_1^\perp}(v_2)$.

2.1.3.3. *Orthogonality and algebraic duality.* The *dual* or *polar* lattice $\Lambda^\vee$ of a lattice Λ is the set $\mathrm{Hom}(\Lambda, \mathbf{Z})$ of $\mathbf{Z}$-module homomorphisms from Λ to $\mathbf{Z}$ with the pointwise module structure, endowed with the dual norm defined by:

$$\|\varphi\| = \inf_{v \in \Lambda \setminus \{0\}} \frac{\varphi(v)}{\|v\|},$$

for any $\varphi \in \Lambda^\vee$. Equivalently the dual lattice can be defined as the module $\{x \in \Lambda_{\mathbf{R}} \mid \forall v \in \Lambda, \langle x, v \rangle \in \mathbf{Z}\}$ endowed directly with $\|\cdot\|_\Lambda$, by representation of linear forms over the finite dimensional space $\Lambda_{\mathbf{R}}$.

Let $\Lambda' \subset \Lambda$ be a pure sublattice. Define its *orthogonal* in $\Lambda_{\mathbf{R}}$ to be the sublattice $\Lambda'^\perp = \{x \in \Lambda^\vee : \langle x, \Lambda' \rangle = 0\}$ of $\Lambda^\vee$. By the definition of the Euclidean structure on the quotient, it is isometric to $\left(\Lambda / \Lambda'\right)^\vee$, so that we can identify the two lattices.

2.1.4 Numerical invariants attached to a lattice

In the following of this section, let us fix Λ a lattice of rank d.

2.1.4.1. *Volume and slope.* A classical numerical invariant attached to Λ is its covolume $\mathrm{covol}(\Lambda)$ defined as the volume[2] of a cell $\mathcal{P}(\mathcal{B})$, also called *a fundamental parallelotope*, for any basis $\mathcal{B} = (v_1, \ldots, v_d)$, where:

$$\mathcal{P}(\mathcal{B}) = \left\{ \sum_{i=1}^{d} x_i v_i \,|\, 0 \leq x_i < 1 \right\}$$

is the parallelotope spanned by the vectors of $\mathcal{B}$ in $\Lambda_{\mathbf{R}}$. Figure 2 depicts the situation for a rank-two lattice. This covolume is independent of the choice

2 The volume is taken for the Lebesgue measure on $\Lambda_{\mathbf{R}}$ normalized so that the Gaussian kernel $x \mapsto e^{\frac{-\|x\|^2}{2}}$ integrates over $\Lambda_{\mathbf{R}}$ to $\sqrt{2\pi}^d$.

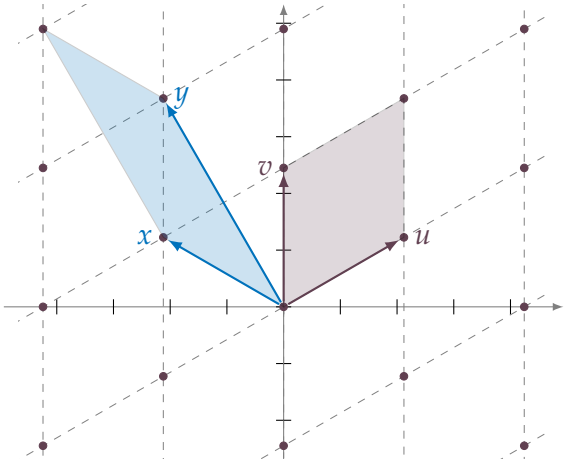


Figure 2: Plane lattice with two bases represented $B = (u, v)$ and $C = (x, y)$ as well as the corresponding parallelotope $\mathcal{P}(B)$ et $\mathcal{P}(C)$. The area of these two bodies is of course the same.

of the basis as being the norm of a basis of the extremal exterior power module $\bigwedge^{\mathrm{rk}\,\Lambda}\Lambda$, of rank 1. Hence, it can effectively be computed from the inner products of the vectors of $\mathcal{B}$:

$$\mathrm{covol}(\Lambda) = \left[\det\left(\langle v_i, v_j\rangle\right)_{1\leq i\leq j\leq d}\right]^{\frac{1}{2}},$$

for a basis $(v_1, \ldots, v_d)$ of Λ. The covolume gives a measure of the density of the lattice in the sense that $\frac{1}{\mathrm{covol}\,\Lambda}$ is the average number of points of Λ per unit volume of $\Lambda_{\mathbf{R}}$, and this intuition is formalized by the following lemma:

Lemma 2.1.3. *Let Λ a rank-d lattice. Then we have:*

$$\frac{1}{\mathrm{covol}\,\Lambda} = \lim_{r\to\infty} \frac{\Lambda \cap \left[-\frac{r}{2}, \frac{r}{2}\right]^d}{r^d}.$$

Proof. Let us fix a basis $\mathcal{B}$ of Λ. Then the open parallelotope $\mathcal{P}(\mathcal{B})$ forms a tiling of $\Lambda_{\mathbf{R}}$ by translation by the elements of Λ: $\bigcup_{v\in\Lambda} \mathcal{P}(\mathcal{B}) + v$. Each of these translations $\mathcal{P}(\mathcal{B}) + v$ intersects Λ only on v by construction. Hence, $\Lambda \cap \left[-\frac{r}{2}, \frac{r}{2}\right]^d = \left|\left\{v \in \Lambda | (\mathcal{P}(\mathcal{B}) + v) \cap \left[-\frac{r}{2}, \frac{r}{2}\right]^d\right\}\right| + o\left(r^d\right)$. But by definition of the volume we have:

$$\left|\left\{v \in \Lambda | (\mathcal{P}(\mathcal{B}) + v) \cap \left[-\frac{r}{2}, \frac{r}{2}\right]^d\right\}\right| = \frac{\mathrm{Vol}\left(\left[-\frac{r}{2}, \frac{r}{2}\right]^d\right)}{\mathrm{Vol}(\mathcal{P}(\mathcal{B}))} + o\left(r^d\right),$$

finishing the proof. ∎

Conversely, and this time not asymptotically, we can prove that a symmetric set which is sufficiently large always contains a lattice point. This is Minkowski's first theorem:

Theorem 2.1.1. *For any lattice Λ of rank d, and any symmetric convex body $K \subset \Lambda_{\mathbf{R}}$ with $\mathrm{Vol}(K/2) > \mathrm{covol}(\Lambda)$, then K contains a non-zero lattice vector, where $K/2 = \{v/2 \,|\, v \in K\}$.*

Proof. Let us suppose that there exists two lattice vectors $v, v' \in \Lambda$ such that the bodies $K/2 + v$ and $K/2 + v'$ have non-empty intersection and let y be in this intersection. Then $v - v'$ lies in the intersection $K \cap \Lambda$. Indeed, $y - v$ and $y - v'$ lies in $K/2$ by construction and so $\frac{v-v'}{2} = \frac{y-v}{2} + \frac{y-v'}{2} \in K/2$ by convexity. To conclude the proof let us suppose that there exists no such pair v, v'. Then this means that for any $r > 0$: $S(r) = \bigcup_{v \in \Lambda \cap [-r/2, r/2]^d} (K/2 + v)$ is a disjoint union so that we have:

$$\mathrm{Vol}(S(r)) = \left|\Lambda \cap [-r/2, r/2]^d\right| \cdot \mathrm{Vol}(K/2) > \left|\Lambda \cap [-r/2, r/2]^d\right| \cdot \mathrm{covol}\,\Lambda$$

by hypothesis. But clearly $\mathrm{Vol}(S(r)) \leq r^d + \mathrm{o}(r^d)$ since $S(r) \subset K/2 + [-r/2, r/2]^d$. This asymptotically contradicts Lemma 2.1.3. ∎

Definition 2.1.7 (Degree). *Let Λ be a lattice. Its* degree *is defined the logarithm of its covolume* $\deg \Lambda = \log \mathrm{covol}\,\Lambda$.

Remark. *In the previous definition, we deliberately take the degree to be the opposite of the regular Arakelov degree of a vector bundle, as it eases some computations.*

The degree is compatible with the monoidal constructions over lattices. Namely:

Lemma 2.1.4. *Let Λ, Λ' two lattices. Then:*

1. $\deg(\Lambda \otimes_{\mathbf{Z}} \Lambda') = \mathrm{rk}\,\Lambda' \deg \Lambda + \mathrm{rk}\,\Lambda' \deg \Lambda'$
2. $\deg(\Lambda \oplus \Lambda') = \deg \Lambda + \deg \Lambda'$.

Proof. Please refer to Appendix 1. ∎

And as such we have the following relation with the covolume of the lattice and of its dual:

Lemma 2.1.5. *Let Λ a lattice then:* $\deg \Lambda = -\deg \Lambda^\vee$.

Proof. Please refer to Appendix 1. ∎

2.1.4.2. *Successive minima.* We have seen that the covolume gives a quantification of the *size* of a given lattice, or equivalently of its density. To be more precise on the actual length of the vectors in the lattice, we can construct by discreteness of Λ a sequence of numerical invariants encoding the norm of small vectors of the lattice:

$$\lambda_i(\Lambda) = \min\{\lambda > 0 \mid \exists\, v_1, \ldots, v_i \in \Lambda \cap B_0(\lambda),\ \mathrm{rk}(v_1, \ldots v_i) = i\},$$

where $B_0(\lambda)$ is the closed ball of radius λ, centered on 0 in $\Lambda_{\mathbf{R}}$. These are the *successive minima* of the lattice Λ.

A bound on the length of the first vector is given by the normalized volume of the lattice up to a factor $\sqrt{d}$. It follows directly from Minkowski's first theorem when taking a d-ball for body K:

Corollary 2.1.1 (Minkowski's first theorem for λ_1)**.** *For any lattice Λ of rank d,*

$$\lambda_1(\Lambda) \leq \sqrt{d}(\operatorname{covol}\Lambda)^{\frac{1}{d}}.$$

The right-hand side of this equation only depends on the dimension and of the covolume of Λ. Therefore, we can wonder what is the best upper-bound γ_d, such that for all lattice Λ of rank d there exists non-zero lattice vector such that we have $\|v\| \leq \gamma_d(\operatorname{covol}\Lambda)^{\frac{1}{d}}$. This corresponds to evaluate the constant:

$$\sqrt{\gamma_d} = \max_{\Lambda}\left[\frac{\lambda_1(\Lambda)}{(\operatorname{covol}\Lambda)^{\frac{1}{d}}}\right],$$

where the minimum is taken on the set of real lattices of rank d. This quantity is called the "d-dimensional Hermite's constant". Of course, Minkowski's theorem gives an upper bound on this constant. Known refinement gives the estimate:

$$\left(\frac{2\Gamma\left(\frac{n}{2}+1\right)\zeta(d)}{\pi^{n/2}r^n}\right)^{\frac{2}{d}} \leq \gamma_d \leq \frac{1.744d}{2\pi e} + o(d).$$

by using Minkowski-Hlawka theorem [80] for the lower bound and analytical arguments for the upper bound (see for instance [37]), with Γ and ζ being respectively the classical Gamma and Zeta function.

Minkowski's first theorem considers the shortest nonzero vector, i.e. the first successive minimum λ_1. A strengthening of the bound is given by what is known as Minkowski's second theorem. Instead of considering the first minimum, this bound considers the geometric mean of all λ_i (which is at least λ_1^d).

Theorem 2.1.2 (Minkowski's second theorem)**.** *The successive minima satisfy*

$$\lambda_1\lambda_2\cdots\lambda_d \leq \gamma_d^{\frac{d}{2}}\operatorname{covol}(\Lambda).$$

Proof. See for instance [8]. ∎

Now that we proved that such small vectors exist in the lattice, we can wonder what is the algorithmic complexity of exhibiting them, and more generally what is the computational cost of problems related to lattices.

2.2 COMPLEXITY OF LATTICES PROBLEMS

We present here some of the basic computational problems that appear naturally when dealing with lattices. For the moment we deliberately choose not to talk of the well-known LWE problem, introduced in the field of lattice-based cryptography. We delay the introduction of this problem to the final part of the manuscript. For the rest of this section, as we are interested in complexity questions, we will consider *rational lattices*, that is lattices which can be described by a basis $v_1, \ldots, v_d \in \mathbf{Q}^n$ of $\mathbf{R}^n$. Without loss of generality we can multiply such lattices by the common denominator of all the coefficients so that we look at the properties of rank-d lattice represented by a matrix of $\mathbf{Z}^{n\times d}$.

2.2.1 Normal form of a lattice

Definition 2.2.1. *An n-by-d matrix A with integer entries has a (column) Hermite normal form H if there is a square unimodular matrix U where $H = AU$ and H satisfies:*

1. *H is lower triangular, and any columns of zeros are located on the right:*
2. *The leading coefficient (the first nonzero entry from the top, also called the* pivot*) of a nonzero column is always strictly below the leading coefficient of the column before it.*
3. *The pivot of any column is positive.*
4. *The elements to the right of pivots are zero and elements to the left of pivots are non-negative and strictly smaller than the pivot.*

Lemma 2.2.1. *Any rational lattice $\Lambda \subset \mathbf{R}^n$ admits a basis in Hermite normal form.*

Proof. We will prove that we can transform any matrix A with integral coefficients into its HNF form by sequences of elementary transformations. More precisely, we construct a sequence A_k of matrices for $k > 0$, such that $A_0 = A$ and

$$A_k = \begin{pmatrix} H_k & 0 \\ C_k & D_k \end{pmatrix},$$

with H_k in Hermite normal form. The matrix A_{k+1} is constructed as follows. Let $d_1, \ldots, d_{n-k}$ the coefficients of the first row of D_k, which can be supposed all non-negative by permuting and taking the opposite of the columns of D_k. Then, by simulating the Euclidean algorithm between the coefficients d_i and d_j with elementary operations on the columns, we can suppose that all the d_i are zero except one which is g, the GCD of $d_1, \ldots, d_{n-k}$. Up to permutation it can be put in the first position of the row. It then remains to satisfy that the coefficients in the first row of C_k are non-negative and smaller than g. To do so, we substract $\lfloor c_i/g \rceil$ times the $(k+1)$-th column to the i-th column in B_k, for each index $i = 1, 2, \ldots, k-1$, which does not affect the entries of H_k. ∎

The algorithm hinted in this proof, which is very close to the method used by Hermite, is however not polynomial in the size of the input. Indeed, the entries may grow exponentially during its execution. It was not until 1979 that an algorithm for computing the Hermite normal form that ran in strongly polynomial time was first developed by Kannan et Bachem in [93]. The fastest asymptotic variant for the computation of the HNF is the one of [156] of Storjohann and Labahn, whose running time is a $O(m^\omega n B)$ for a $m \times n$ matrix whose coefficients represented with B bits.

An interesting consequence of the Hermite normal form is that the covolumes of the successive sublattices induced by the basis are necessarily increasing.

Lemma 2.2.2. *Given $(v_1, \dots, v_d)$ a basis in Hermite normal form of a d-dimensional integer lattice $\Lambda \subset \mathbf{Z}^d$, then, for any $1 \leq i < d$,*

$$\operatorname{covol}[v_1\mathbf{Z} \oplus \cdots \oplus v_i\mathbf{Z}] \leq \operatorname{covol}[v_1\mathbf{Z} \oplus \cdots \oplus v_{i+1}\mathbf{Z}].$$

In particular, for any sublattice Λ' generated by the m first vectors $v_1, \dots, v_m$:

$$\operatorname{covol} \Lambda' \leq \operatorname{covol} \Lambda.$$

Proof. Let $(v_1, \dots, v_d)$ a basis in Hermite normal form of $\Lambda \subset \mathbf{Z}^d$. Then, since its matrix M in the canonical basis of $\mathbf{Z}^d$ is triangular with non-negative coefficients, the sequence $(m_{1,1}, m_{1,1} \times m_{2,2}, \dots, m_{1,1} \times \cdots \times m_{n,n})$ of successive products is increasing. It then suffices to remark that its k-th term corresponds to the determinant of the lattice generated by the k-th first vectors $v_1, \dots, v_k$. ∎

2.2.2 Shortest and closest vectors problems

2.2.2.1. *Exact problems and their hardness.* We now define two classical complexity problems, related to the geometry of lattices:

Problem (SVP)**.** *Let Λ be a lattice, the* shortest vector problem *(SVP) is defined as the search problem:*

> ***Input:*** *A basis of Λ.*
>
> ***Output:*** *A vector $v \in \Lambda$ such that $\|v\| = \lambda_1(\Lambda)$.*

Problem (CVP)**.** *Let Λ be a lattice, the* closest vector problem *(CVP) is defined as the search problem:*

> ***Input:*** *A basis of Λ, a vector $x \in \Lambda \otimes \mathbf{Q}$.*
>
> ***Output:*** *A vector $v \in \Lambda$ such that $\|v - x\|$ is minimal over Λ.*

Remark. *We only describe the so-called* search *variants, since this is the problems we will be interested in and let their (equivalents under mild conditions)* decisional *counterparts aside.*

The closest vector problem is *NP-hard*, which means that a polynomial time algorithm for this problem would give a polynomial time algorithm for any problem in the class NP[3]. The reduction is pretty straightforward, as being a reduction to the *subset-sum problem.*

The shortest vector problem is only known to be *NP-hard for randomized reductions* (see for instance [94]), which means that a polynomial time algorithm for this problem would give a randomized polynomial time algorithm for any problem in the class NP.

3 NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.

2.2.2.2. *Approximate variants.* Since these problems are hard, one can relax them into approximate versions: we do not seek exactly for the shortest or closest vector but for any vector inside a ball of given radius, expressed as a factor of the Hermite constant or of the covering radius.

Problem (γ-SVP)**.** *Let Λ be a lattice, the* approximate shortest vector problem with approximation factor γ *(γ-SVP) is defined as the search problem:*

Input: *A basis of Λ.*

Output: *A vector $v \in \Lambda$ such that $\|v\| \leq \gamma\lambda_1(\Lambda)$.*

Problem (γ-CVP)**.** *Let Λ be a lattice, the* approximate closest vector problem with approximation factor γ *(γ-CVP) is defined as the search problem:*

Input: *A basis of Λ, a vector $x \in \Lambda \otimes \mathbf{Q}$.*

Output: *A vector $v \in \Lambda$ such that*

$$\|v - x\| \leq \gamma \min_{v \in \Lambda} \|v - x\|.$$

Of course, the hardness of the γ-SVP problem depends on γ. Up to approximation factors of $2^{\frac{1}{2}(\log d)-\epsilon}$ for $\epsilon > 0$, the problem is NP-hard, but $\sqrt{d}$-SVP has been proved to be in NP$\cap$co-NP, making it unlikely to be NP-hard. For high approximations factors, the situation is elucidated. For instance, we demonstrate at the end of this chapter, that the best known polynomial-time algorithm achieves an approximation factor of $2^{O\left(\frac{n \log\log n}{\log n}\right)}$.

Remark. *A similar problem to the γ-SVP, sometimes used to compare lattice reduction algorithm,is the so-called γ-Hermite-SVP problem, which is defined as:*

Problem (Hermite γ-SVP)**.** *Let Λ be a lattice, the γ-Hermite shortest vector problem is defined as:*

Input: *A basis of Λ.*

Output: *A vector $v \in \Lambda$ such that: $\|v\| \leq \gamma \operatorname{covol} \Lambda^{\frac{1}{n}}$.*

As such, we can try to look for tractable approximations of solutions of these problems for arbitrary lattices. The systematic treatment of this question is the field of algorithmic reduction theory, which aims at designing algorithms which compute a somewhat short basis of a lattice given as input. We choose to start this discussion by the simplest possible, but non-trivial, case: the reduction of rank-2 lattices.

2.3 LAGRANGE-GAUSS' REDUCTION

Let Λ be a two-dimensional lattice, given as a basis (u, v). Without loss of generality, we can suppose that $\|u\| \leq \|v\|$, by rearranging the order of

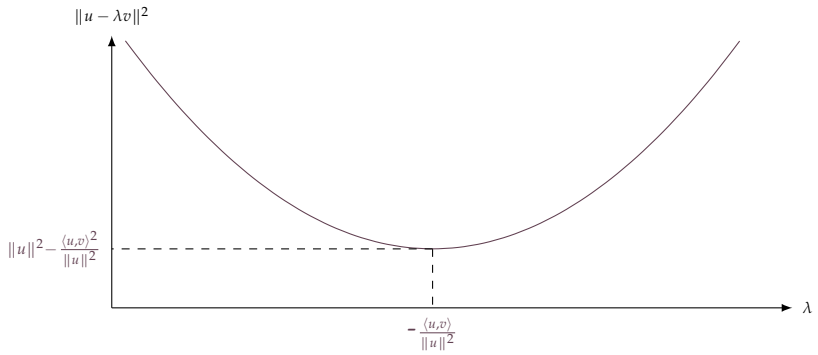


Figure 3: Norm of the coset elements $v - \lambda u$

these vectors. Now that the first vector of the basis is the smallest among the two, we try to reduce the norm of the second one. Explicitly we want to find the shortest vector v' such that (u, v') is a basis of Λ. This condition means that v' and v must be equal up to sign in the quotient $\Lambda/u\mathbf{Z}$, since they must have the same degree $\deg \Lambda - \deg u\mathbf{Z}$ in this quotient lattice of rank 1. Hence this means that $v' \in v + u\mathbf{Z}$. The problem is now reduced to finding the shortest vector in this coset.

2.3.1 Vector of minimal norm in the coset

To find such a vector, let then explicitly write the norm of any vector in $v + \mathbf{Z}u$. For any parameter $\lambda \in \mathbf{R}$, we have by the parallelogram law and bilinearity of the norm:

$$\|v - \lambda u\|^2 = \|v\|^2 + \lambda^2\|u\|^2 - 2\lambda\langle v, u\rangle \tag{2.1}$$

The right hand side of Equation 2.1 is a degree two real polynomial in λ, of discriminant $4(\langle u, v\rangle^2 - \|u\|^2\|v\|^2) = -4 \operatorname{covol} \Lambda < 0$. Henceforth it has no real roots, and its (unique) global minimum is attained at $\lambda_{min} = -\frac{\langle u,v\rangle}{\|u\|^2}$. Remark now that the graph of function $(\lambda \mapsto \|v - \lambda u + \lambda_{min} u\|^2)$ is even, so that its minimum over $\mathbf{Z}$, is attained at the closest integral point to λ_{min} that is: $\lfloor \lambda_{min} \rceil$.

2.3.2 Iterating the reduction

Once such a small representant v' of $\Lambda/\mathbf{Z}u$ is found, we obtain a new basis of (u, v') of Λ. If $\|v'\| < \|u\|$ we can start this whole reduction once again. Looping this process eventually terminates, since the size of the first vectors shrinks at each step and is a non-negative integer. The whole algorithm is described in an iterative manner in Algorithm 7.

Algorithm 7 — Gauss reduction

Input : (u, v) a basis of a two-dimensional lattice Λ
Output : A reduced basis (u, v) of Λ

1 **if** $\|v\| < \|u\|$ **then return Gauss**(v, u)
2 $v' \leftarrow v - \left\lfloor \frac{\langle u,v\rangle}{\|u\|^2} \right\rceil u$
3 **if** $\|v'\| < \|v\|$ **then return Gauss**(u, v')
4 **else return** (u, v)

2.3.3 On the properties of reduced bases

Let (u, v) be a reduced basis of a lattice Λ. By construction, these vectors satisfy

$$\|u\| \leq \|v\|$$

and:

$$\left\lfloor \frac{\langle u, v\rangle}{\|u\|^2} \right\rceil = 0, \qquad \text{i.e} \qquad |\langle u, v\rangle| \leq \frac{\|u\|^2}{2}.$$

From these observations, we can give a more axiomatic definition of a reduced basis:

Definition 2.3.1 (Gauss-reduced basis)**.** *Let Λ be a two dimensional real lattice. A basis (u, v) of Λ is said to be* Gauss-reduced *if it fulfills the two conditions:*

1. $\|u\| \leq \|v\|$
2. $|\langle u, v\rangle| \leq \frac{\|u\|^2}{2}$.

Proposition 2.3.1. *Let (u, v) be a reduced basis of a lattice Λ, then $\|u\| = \lambda_1(\Lambda)$ and $\|v\| = \lambda_2(\Lambda)$.*

Proof. Let (u, v) be a reduced basis of a lattice Λ. Remark that by definition of the reduction, we have that $\|v\| \leq \|v \pm u\|$ and by the analysis of Equation 2.1, we have more generally $\|v\| \leq \|v + \lambda u\|$ for any integer $\lambda \neq 0$.

Let now $x = \alpha u + \beta v$ be a generic point of Λ. Clearly if $\beta = 0$ we have $\|x\| \geq \|u\|$. We can now suppose without of generality that $\beta > 0$ and set $\alpha = \kappa\beta + \rho$ with $0 \leq \rho < \beta$ to be the Euclidean division of α by β. Then we have by reverse triangular inequality:

$$\begin{aligned} \|\alpha u + \beta v\| &\geq \beta\|v + \kappa u\| - \rho\|u\| \\ &= (\beta - \rho)\|v + \kappa u\| + \rho(\|v + \kappa u\| - \|u\|) \\ &\geq \|v + \kappa u\| \geq \|v\| \geq \|u\|, \end{aligned}$$

as $\|v + \kappa u\| - \|u\| \geq 0$ and $\beta - \rho$ is an integer greater than o. This ends the proof. ∎

Using the existence of such reduces bases, we can relate the length of the shortest vector to the global determinants, retrieving Minkovsky's first theorem for lattices of rank 2.

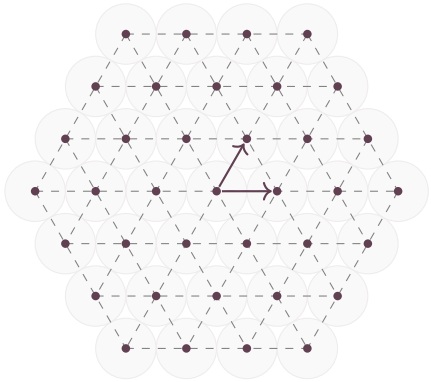
Corollary 2.3.1 (Minkovsky first theorem for planes)**.** *Let Λ be a lattice of rank 2. Then: $\lambda_1(\Lambda) \leq \sqrt{\frac{4}{3}\det\Lambda}$.*

Proof. Let (u,v) be a reduced basis of Λ, then $\langle u,v\rangle^2 \leq \frac{\|u\|^4}{4}$. As such, we have by definition:

$$\det\Lambda^2 = \|u\|^2\|v\|^2 - \langle u,v\rangle^2 \geq \|u\|^2\|v\|^2 - \frac{\|u\|^4}{4} = \frac{3}{4}\|u\|^4.$$

■

Remark. *This bound is optimal, as being reached for the critical lattice of dimension 2: $\left(\frac{1}{2},\frac{\sqrt{3}}{2}\right)\mathbf{Z} \oplus \left(\frac{1}{2},-\frac{\sqrt{3}}{2}\right)\mathbf{Z}$, as represented as follows:*



2.3.4 On the running time of Gauss-reduction process

Theorem 2.3.1. *The number of steps of the Gauss-reduction is $O(B)$, where B is a bound on the number of bits required to represents the basis given as input.*

Proof. Remark first that except for the last iteration of the algorithm, the norm of the first vector of the basis shrinks by a factor of at least $\sqrt{3}^{-1}$. Indeed let (u,v) be the current state of the basis at any step except the final one and (u,v') the basis one step after. Remark that:

$$|\langle u',v'\rangle| = |\langle v,u'\rangle| = \left|\left\lfloor\frac{\langle u,v\rangle}{\|u\|^2}\right\rceil - \frac{\langle u,v\rangle}{\|u\|^2}\right|\|u\|^2 \leq \frac{1}{2}\|u\|^2.$$

Suppose that $\|u'\|^2 \geq \|u\|/3$, then we would have:

$$|\langle u',v'\rangle| \leq \frac{3}{2}\|u'\|^2.$$

In this case, during the next iteration of the algorithm, line 6 of Algorithm 7 makes appear $u' \pm v'$. If this vector appears to be smaller than u' then we would have already computed it differently in the current iteration. Hence, the next step is the final iteration. We can conclude since the norm of the first vector is initially bounded by 2^B. ■

Remark. *On a historical note, the reduction algorithm appears as natural generalization of Euclid's GCD algorithm in Lagrange's letters of 1773. It was also described later by Gauss in 1801.*

Now that we can reduce lattices in rank 2 and reach the first minima of such lattices, we might be interested in efficiently reducing lattices in arbitrary dimension, that is finding a *polynomial time* reduction procedure.

2.4 TOWARDS POLYNOMIAL TIME REDUCTION OF LATTICES

In the following of this section, let us fix Λ a lattice of rank d. Suppose that we are given a basis $\mathcal{B} = (v_1, \ldots, v_d)$ of Λ. Let us denote by Λ_i the sublattice of rank i generated by the first $i - th$ vectors that is: $\Lambda_i = v_1\mathbf{Z} \oplus \cdots \oplus v_i\mathbf{Z}$ and by $\pi_i : \Lambda \to \Lambda/\Lambda_i$ the corresponding canonical projection[4]. Such a choice yields a *filtration* of $\mathbf{Z}$-modules:

$$\mathcal{F} : \{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_d = \Lambda.$$

We define the corresponding *profile* $\mu\mathcal{B}$ of the basis $\mathcal{B}$ (or of the filtration $\mathcal{F}$) to be the sequence of degrees:

$$\mu\mathcal{B} = (\deg \Lambda_1, \deg \Lambda_2, \ldots, \deg \Lambda).$$

Like for the Gauss' reduction algorithm, we aim at finding vectors of Λ which are the shortest possible, that is trying to solve the SVP problem for the lattice Λ. Of course, by the NP-hardness result of Paragraph 2.2.2.1, such a problem is hard. Henceforth, we will only look for *reasonably* short vectors. Since we do not know *a priori* how to reduce the lattice Λ as soon as $\operatorname{rk} \Lambda > 2$, we will aim at reducing this problem to a succession of instances of reductions in smaller rank. We proceed by successive *densification* of the filtration $\mathcal{F}$. This process is based on the following simple idea. Suppose that the first vector v_1 is small—i.e., that the first element Λ_1 has small degree—. Now, if we try to make the degree of Λ_2 as small as possible, among all sublattices of rank two containing Λ_1, then by Hadamard's inequality v_2 is also small (even though not necessarily the smallest possible). Then, similarly, since we want the third vector to be small as well, we try to minimize the degree of Λ_3, and so on for the subsequent elements of $\mathcal{F}$. All in all, we aim at making the coefficients of the profile of the filtration as small as possible, that is, making each of the Λ_i as dense as we can. This justifies coining the term *densification* of a basis.

We start be describing such a reduction process abstractly, using only a filtration, and becomes gradually closer to vectors and practical considerations in order design and analyze an actual algorithm.

4 We also use this same notation π_i to denote the orthogonal projection over the space $\Lambda_i^{\perp}$ in $\Lambda_{\mathbf{R}}$ since the projection of $\pi_i(\Lambda)$ is isometric to the quotient Λ/Λ_i.

2.4.1 From Gauss' reduction to LLL reduction

2.4.1.1. *An action on filtration.* The Gauss reduction algorithm acts naturally on (complete) filtrations. Let us explicitly write this action. At index $1 \leq i \leq d$, we can construct a rank 2 quotient lattice:

$$\Lambda^* = {}^{\Lambda_{i+1}}\!/_{\Lambda_{i-1}}.$$

Then, by Corollary 2.3.1, Gauss's reduction on this lattice finds a reduced basis of Λ^*, that is a complete filtration $\{0\} \subset \Lambda_1^* \subset \Lambda^*$, satisfying

$$2 \deg \Lambda_1^* \leq \deg \Lambda^* + \log\left(\frac{4}{3}\right).$$

We can now *lift* it into a subfiltration $\Lambda_{i-1} = \Lambda_i' \subset \Lambda_{i+1}$, such the quotient filtration by Λ_{i-1} is equal to the reduced filtration:

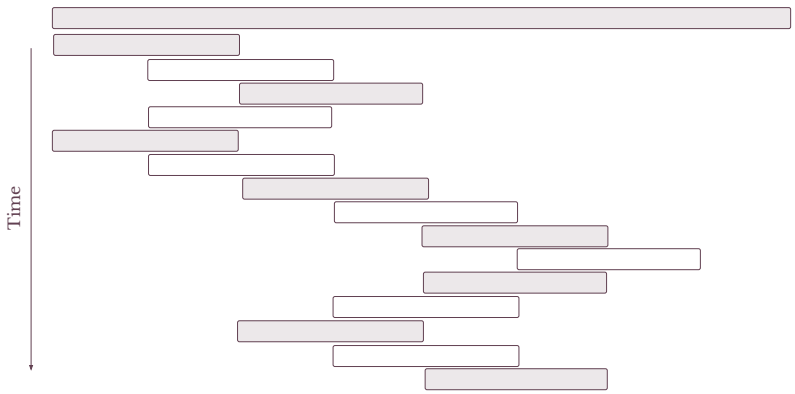
$$\begin{array}{ccccc} {}^{\Lambda_{i-1}}\!/_{\Lambda_{i-1}} & \subset & {}^{\Lambda_i'}\!/_{\Lambda_{i-1}} & \subset & {}^{\Lambda_{i+1}}\!/_{\Lambda_{i-1}} \\ \| & & \| & & \| \\ \{0\} & \subset & \Lambda_1^* & \subset & \Lambda^* \end{array}.$$

This lift can be done by using the restriction $(\pi_i)_{|\Lambda_{i+1}}$ of the canonical projection onto the quotient ${}^{\Lambda_i'}\!/_{\Lambda_{i-1}}$. It suffices to set $\Lambda_i' = v\mathbf{Z} \oplus \Lambda_{i-1}$ for a representative $v + \Lambda_{i-1}'$ of the rank one lattice Λ_1^*.

2.4.2 Densification of a basis

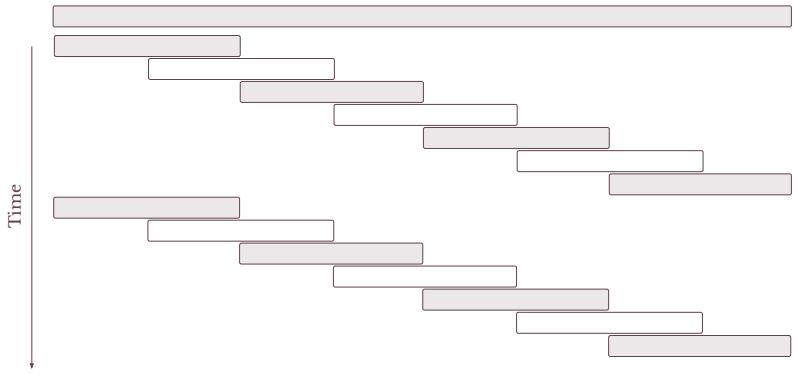
Applying Gauss' reduction to filtration *locally densifies* it, as it reduces the degree of the quotients ${}^{\Lambda_i}\!/_{\Lambda_{i-1}}$. To transform this local process to a global one, at least two possible iterative strategies opens from this observation:

Incremental progression: We try to densify the filtration starting from Λ_1 and incrementally progress through the filtration. If a modification is done for the sublattice Λ_i we make a step back and start again at position $i - 1$ in the filtration. Graphically, a generic execution might looks like:



where we progress along the basis but with local back and forth (a small block indicates where the local reduction is done in the filtration). When we eventually reach the end of the filtration, no local reduction is able to alter the filtration and we can stop.

Ironing out strategy: We perform a local reduction at each index and start again when the end of the filtration is reached. We then start all over again from the beginning until no change are made by this process. Graphically, the first steps of a generic execution might looks like:



We first look at the first strategy as it recovers the historical LLL algorithm and discuss the difference with the second one thereafter. At the end of the process, no local progress can be done on the filtration, and as such, by construction, the sequence of degrees satisfies the relation:

$$2\deg\left(\Lambda_i/\Lambda_{i-1}\right) \leq \deg\left(\Lambda_{i+1}/\Lambda_{i-1}\right) + \frac{1}{2}\log\left(\frac{4}{3}\right),$$

for any $1 \leq i \leq d-1$, yielding by additivity of the degree over short exact sequences:

$$\deg \Lambda_i - \deg \Lambda_{i-1} \leq \deg \Lambda_{i+1} - \deg \Lambda_i + \frac{1}{2}\log\left(\frac{4}{3}\right),$$

with the convention that $\deg \Lambda_0 = 0$. Summing these relations for all $1 \leq i \leq k$ gives for any $j > k$:

$$\deg \Lambda_k \leq \frac{1}{2}\log\left(\frac{4}{3}\right) + k(\deg \Lambda_{j+1} - \deg \Lambda_j).$$

Thus, by summing d times and separating indices below k and beyond k we have:

$$\begin{aligned} d\deg \Lambda_k &\leq k\deg \Lambda_k + \sum_{j=k+1}^{d}\left((\deg \Lambda_{j+1} - \deg \Lambda_j) + \frac{k(2j-k-1)}{4}\log\left(\frac{4}{3}\right)\right) \\ &\leq \deg \Lambda_d + \frac{d(d-k)k}{4}\log\left(\frac{4}{3}\right). \end{aligned}$$

In particular we have:

$$\deg \Lambda_1 \leq \frac{d-1}{4} \log\left(\frac{4}{3}\right) + \frac{1}{d} \deg \Lambda_d$$

Remark. *In substance this abstract procedure is similar to the technique used by Hermite in 1845, proving the so-called Hermite's inequality:*

$$\gamma_d \leq \gamma_2^{d-1}.$$

We can directly recover this inequality from our result, by remarking that:

- *By definition of the degree, we have* $\log \lambda_1(\Lambda) \leq \deg \Lambda_1$,
- *By Corollary 2.3.1,* $\gamma_2 = \sqrt{\frac{4}{3}}$.

So that:

$$\sqrt{\gamma_d} \leq \frac{\lambda_1(\Lambda)}{\operatorname{covol} \Lambda} \leq \gamma_2^{\frac{d-1}{2}}.$$

2.4.3 Towards an actual algorithm

In order to transform this abstract procedure on filtrations in an actual algorithm we need to explicitly describe the lifting procedure above-mentionned. Using the same notations as in Section 2.4.1, this question translates to vectors in choosing an element of the coset $v + \Lambda'_{i-1}$.

2.4.3.1. *Finding a small representative in a coset.* If we move back from the filtration point of view to the vector one, we might want to take the shortest possible vector of this coset. However, as stated in Paragraph 2.2.2.1, this is a CVP instance in dimension $\operatorname{rk} \Lambda_{i-1} = i - 1$, which is *a priori* hard. However if we *relax* the condition of finding the closest vector to v in finding a vector relatively close to it, the problem becomes much simpler, as we will break this search in successive instances of reductions of two vectors.

Remark that we can use the subfiltration $\Lambda_0 \subset \cdots \subset \Lambda_{i-1}$ to greedily find such a representation. Suppose that this latter filtration arised from the basis $(v_1, \ldots, v_{i-1})$. Then, we can start by reducing the vector v *with* the vector v_{i-1}, that is looking for a small representative of the class $v_i + \mathbf{Z}v_{i-1}$. But as at this point, since we do not have reduced v against the other vectors $v_1, \ldots, v_{i-2}$, we do want to make vanish the contribution of these first $i-2$ vectors when trying to reduce v with v_{i-1}. Informally we want to reduce these vectors *independently* of all the possible linear combinations with elements in Λ_{i-2}. This means performing the reduction of v *modulo* the whole additive action of the lattice Λ_{i-2}. This simply translates as performing the reduction step *inside the quotient* Λ/Λ_{i-2}, and then lifting the resulting vector in a new vector $v'_i \in v_i + v_{i-1}\mathbf{Z} \subseteq v + \Lambda_{i-1}$. The same

quadratic minimization as the one performed in Section 2.3 reveals that the minimum norm is reached for

$$\left(v + \left\lceil \frac{\langle v, \pi_{i-1}(v_{i-1})\rangle}{\|\pi_{i-1}(v_{i-1})\|^2} \right\rfloor v_{i-1}\right) + \Lambda_{i-2} \in {}^{\Lambda}\!/_{\Lambda_{i-2}},$$

so that the lifted vector is $v' = v + \left\lceil \frac{\langle v, \pi_{i-1}(v_{i-1})\rangle}{\|\pi_{i-1}(v_{i-1})\|^2} \right\rfloor v_{i-1}$.

Once this operation is performed, we can start to go back in the basis and reduce the newly found v' with v_{i-2}. As before, we do perform this reduction modulo Λ_{i-3}, that is reducing the projection of v' by v_{i-2} inside the quotient ${}^{(\Lambda_{i-2} \oplus v\mathbf{Z})}\!/_{\Lambda_{i-3}}$. We can then perform this reduction up to getting down to to v_1.

This recursive reduction can be simply made iterative and gives the procedure called *weak* reduction, or *size-reduction*, written in pseudo-code in Algorithm 8.

Algorithm 8 — Size-reduction

Input : $(v_1, \dots, v_d)$ a family of vectors
Output : A size-reduced family $(v_1, \dots, v_d)$.

```
for k = 2 to n do
    for j = k − 1 downto 1 do
        v_k ← v_k − ⌈⟨v_k, π_j(v_j)⟩ / ‖π_j(v_j)‖²⌋ · v_j
    end for
end for
return (v_1, ..., v_d)
```

2.4.3.2. *An iterative reduction process.* If we breaks the steps of the Gauss reduction of the above-described procedure and combine these exchanges with the weak-reduction we get a blueprint of reduction, exposed in pseudo-code in Algorithm 9, with the convention that v_0 is the zero vector.

Algorithm 9 — Prototype of reduction

Input : $(v_1, \dots, v_d)$ a basis of a rank d lattice Λ
Output : A *reduced basis* of Λ

```
while progress is done do
    Size-reduce(v_1, ..., v_d)
    if ∃i, deg(v_0, ··· , v_{i−1}, v_{i+1}) < deg(v_0, ··· , v_{i−1}, v_i) then
        Swap v_i and v_{i+1}
    end if
end while
return (v_1, ..., v_d)
```

Let us optimize the procedure a bit. First, we can order the exchange form the left part of the basis to its right and keeping track of the latest exchange to avoid performing a full size-reduction. Then remark that an exchange between the two vectors is then performed if

$$\deg(\Lambda_{i-1} \oplus v_{i+1}\mathbf{Z}) < \operatorname{covol}(\Lambda_{i-1} \oplus v_i\mathbf{Z}),$$

that is if $\|v_{i+1} + \Lambda_{i-1}\| < \|v_i + \Lambda_{i-1}\|$, since by definition of the quotient we have $\deg(\Lambda' \oplus x\mathbf{Z}) = \deg(\Lambda') + \log\|x + \Lambda'\|$ for any strict sublattice Λ' and vector $x \notin \Lambda'$. Expanding the inequality using the identification of the quotient lattice with the orthogonal projection and relaxing[5] the strict inequality by a continuous parameter $0 < \delta < 1$ yields the condition:

$$\delta\|\pi_{i-1}(v_{i-1})\|^2 \leq \|\pi_i(v_i)\|^2 + \langle v_i, \pi_{i-1}(v_{i-1})\rangle^2 \|\pi_{i-1}(v_{i-1})\|^2$$

The whole iteration of this process is described in an high-level manner in Algorithm 10. This is the *Lenstra-Lenstra-Lovász* algorithm, as presented in [109].

Algorithm 10 — Textbook LLL reduction

Input : $(v_1, \ldots, v_d)$ a basis of a rank d lattice Λ
Output : A δ-reduced basis of Λ

$k \leftarrow 2$
Compute the $\pi_i(v_i)$'s with the **Gram-Schmidt** process
while $k \leq d$ **do**
  for $j = k-1$ **downto** 1 **do** $v_k \leftarrow v_k - \left\lceil \frac{\langle v_k, \pi_j(v_j)\rangle}{\|\pi_j(v_j)\|^2} \right\rfloor \cdot v_j$
  if $\delta\|\pi_{k-1}(v_{k-1})\|^2 \leq \|\pi_k(v_k)\|^2 + \langle v_k, \pi_{k-1}(v_{k-1})\rangle^2 \|\pi_{k-1}(v_{k-1})\|^2$ **then**
    $k \leftarrow k+1$
  else
    Swap v_k and v_{k-1}; Update $\pi_k(v_k)$ and $\pi_{k-1}(v_{k-1})$
    $k \leftarrow \max(k-1, 2)$
end while
return $(v_1, \ldots, v_d)$

On the QR-decomposition. Before pursuing this introduction to reduction algorithms, we point out the matrix interpretation of the decomposition induced by the projections π_i. Given an invertible matrix B, representing a basis $\mathcal{B}$ of Λ, we have

$$B = QR$$

by setting $Q = \left[\frac{\pi_i(v_i)}{\|\pi_i(v_i)\|}\right]_{1 \leq i \leq d}$ and $R = \left(\langle \pi_i(v_i), v_j\rangle\right)_{1 \leq i,j \leq d}$. Q is an orthogonal matrix and R an upper triangular one, by definition of the projections. This decomposition is called the QR-decomposition of B.

5 This relaxation may seem artificial, but it is a classical trick in algorithm design to ensure polynomial time for greedy algorithm.

The Gram-Schmidt orthogonalization process (GSO) is an algorithmic way to compute inductively the projections $\pi(v_i)$ and as such the QR-decomposition of a matrix. It is done by setting $\pi_1(v_1) = v_1$ and then for all $1 < i \leq r$,

$$\pi_{i-1}(v_i) = v_i - \sum_{j=1}^{i-1} \frac{\langle v_i, \pi_j(v_j)\rangle}{\langle \pi_j(v_j), \pi_j(v_j)\rangle} v_j.$$

We can actually rewrite this algorithm using matrices by using the QR-decomposition. More precisely we only use the R part of this decomposition to perform the reduction. This simple rewriting, given in Algorithm 11, is the base from which we construct generalized reductions in algebraic contexts in Chapter 5. It also gives an interesting insight on the LLL reduction: since lattice reduction is invariant by orthogonal transformation of the ambient space, by the QR-decomposition we only need to define a reduction for lattices given by triangular matrices. The LLL reduction acts on such matrices by an iterative sequence of local reductions on the diagonal and over-diagonal elements, forming projected sublattices of rank 2 which are themselves triangular. Thus, we actually retrieved the abstract point of view introduced in Section 2.4.1 using filtrations: the R part of the QR-decomposition is nothing else than a matrix encoding of the filtration (indeed recall that flags of vector spaces are in correspondence with upper-triangular matrices). The reduction is indeed a sequence of local reduction (2×2-triangular matrices, i.e., quotient subfiltrations of dimension 2).

Algorithm 11 — R-based LLL reduction

Input : Initial basis $(v_1, \ldots, v_d)$
Result : A δ-LLL-reduced basis

$k \leftarrow 1$
while $k < d$ **do**
  Compute the R part of the QR-decomposition of B
  for $j = k-1$ **downto** 1 **do** $v_k \leftarrow v_k - \lceil R_{k,j} \rfloor \cdot v_j$
  if $\delta \|(R_{k,k}, 0)\|^2 \leq \|(R_{k+1,k}, R_{k+1,k+1})\|^2$ **then**
    $k \leftarrow k+1$
  else
    Swap v_k and v_{k+1}
    $k \leftarrow \max(k-1, 1)$
end while
return $(v_1, \ldots, v_d)$

2.4.4 Properties of reduced bases

Similarly to the exposition of Gauss reduction algorithm in Section 2.3, we say that a basis is reduced if it is invariant under the application of Algo-

rithm 10. By definition, such a basis verifies two sets of conditions. The unimodular transformation of line 4 should act as the identity, that is that

$$\left\lfloor \frac{\langle v_j, \pi_i(v_i)\rangle}{\|\pi_i(v_i)\|^2} \right\rceil = 0$$

for any index i. The second set of conditions arise from the guard of line 4: for any index i, the norm of the $(i+1)$-th vector should be greater than the norm of the i-th vector. All in all, this leads to the following axiomatic definition of the LLL reduceness notion:

Definition 2.4.1 (LLL reduction)**.** *A basis $(v_1, \ldots, v_d)$ of a lattice is said to be δ-LLL-**reduced** for a certain parameter $1/4 < \delta \leq 1$, if the following conditions are satisfied:*

$$\forall i < j, \quad \left|\langle v_j, \pi_i(v_i)\rangle\right| \leq \frac{1}{2}\|\pi_i(v_i)\|^2 \quad \textit{(Size-Reduction condition)} \tag{2.2}$$

$$\forall i, \quad \delta\|\pi_i(v_i)\|^2 \leq \left(\|\pi_{i+1}(v_{i+1})\|^2 + \langle v_{i+1}, \pi_i(v_i)\rangle^2\right) \quad \textit{(Lovász condition)} \tag{2.3}$$

All in all, we can quantify the density of the successive sublattices:

Theorem 2.4.1. *Consider a δ-LLL reduced basis $(v_1, \ldots, v_d)$ of $(\Lambda, \langle\cdot,\cdot\rangle)$. For any $1 \leq k \leq d$, denoting by Λ_k the sublattice spanned by $v_1, \ldots, v_k$, we have:*

$$\operatorname{covol}(\Lambda_k) \leq \left(\frac{1}{\delta - 1/4}\right)^{-\frac{(d-k)k}{4}} \operatorname{covol}(\Lambda)^{\frac{k}{d}}.$$

Proof. The proof is in substance the same one we did with filtrations, this time, taking the relaxation factor δ into account. Using the Lovász condition at index $1 \leq i < d$, we write:

$$\delta\|\pi_i(v_i)\|^2 \leq \|\pi_{i+1}(v_{i+1})\|^2 + \left(\frac{\langle v_{i+1}, \pi_i(v_i)\rangle}{\|\pi_i(v_i)\|^2}\right)^2 \|\pi_i(v_i)\|^2$$

Thanks to the size-reduction condition, this implies:

$$\forall i \in \{1, \ldots, d-1\}, \quad \|\pi_i(v_i)\|^2 \leq (\delta - 1/4)^{-1}\|\pi_{i+1}(v_{i+1})\|^2. \tag{2.4}$$

Let K denote $(\delta - 1/4)^{-1/2}$ and ℓ_i be the norm of the vector $\pi_i(v_i)$. Then, Equation (2.4) becomes:

$$\forall i \in \{1, \ldots, d-1\}, \quad \ell_i \leq K\ell_{i+1}.$$

Recall that $\operatorname{covol}(v_1, \ldots, v_k) = \prod_{i=1}^{k} \ell_i$. This implies that for any $j > k$:

$$\operatorname{covol}(v_1, \ldots, v_k) \leq \prod_{i=1}^{k} K^{j-i}\ell_j = K^{k(2j-k-1)/2} \cdot \ell_j^k.$$

Thus:

$$\begin{aligned}\operatorname{covol}(v_1,\dots,v_k)^d = \left(\prod_{i=1}^{k}\ell_i\right)^d &\le \left(\prod_{i=1}^{k}\ell_i\right)^k \prod_{j=k+1}^{d} K^{k(2j-k-1)/2}\cdot \ell_j^k \\ &\le \left(\prod_{i=1}^{d}\ell_i\right)^k K^{\sum_{j=k+1}^{d} k(2j-k-1)/2} \\ &\le \operatorname{covol}(\Lambda)^k K^{\frac{d(d-k)k}{2}}.\end{aligned}$$

■

2.4.5 Running time analysis

As shown in [109], the LLL algorithm terminates in polynomial time when $\delta < 1$. The proof is two-fold: first, we prove that the total number of swaps is polynomial, so that the number of iterations is polynomial, and then we conclude by proving that the complexity of an iteration of the loop is also polynomial.

Number of swaps. By construction of the LLL reduction, each swap makes a coefficient of the profile decrease by at least $\log\delta$, so that $\|\mu\mathcal{B}\|_1$ decreases by at least $\log\delta$. The quantity $\|\mu\mathcal{B}\|_1$ is the *potential* of the basis $\mathcal{B}$. Since the total number of iterations can be bounded by twice the number of swaps plus the dimension of the lattice, this suffices to conclude that it is bounded by $\mathrm{O}(d^2B)$ where $B = \log\|\mathcal{B}\|_{\max}$ is a bound on the size of the coefficients of the matrix of the initial basis $\mathcal{B}$. But the cost of a loop iteration is of $\mathrm{O}(dn)$ arithmetic operations on *rational* coefficients.

Complexity of one iteration. Let us conclude this complexity analysis by proving that the length of these coefficients is at most as a $\mathrm{O}(dB)$. We start by bounding the size of the integers appearing during the size-reduction process. Denote by M the triangular matrix $\left(\frac{\langle v_i,\pi_j(v_j)\rangle}{\|\pi_j(v_j)\|^2}\right)_{1\le i<j\le d}$, completed with 1's on its diagonal. Then, by denoting by N its inverse, we can express the Gram-Schmidt orthogonalization of the elements of $\mathcal{B}$ as:

$$\pi_i(v_i) = v_i + \sum_{\ell=1}^{i=1} N_{i,\ell}v_\ell,$$

so that for all $1 \le j \le i-1$ we have:

$$0 = \langle v_i, v_j\rangle + \sum_{\ell=1}^{i-1} N_{i,\ell}\langle v_\ell, v_j\rangle,$$

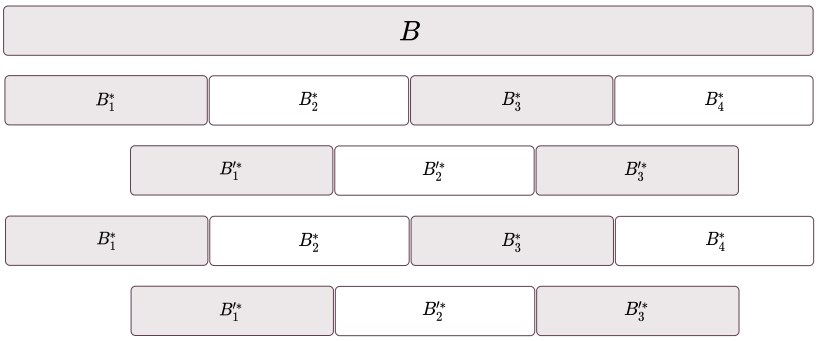
and so:

$$G[1:i-1,1:i-1]\cdot(N_{i,1},\dots,N_{i,i-1})^T = -(\langle v_i,v_1\rangle,\dots,\langle v_i,v_{i-1}\rangle)^T,$$

for G being the Gram matrix of the basis $\mathcal{B}$. Each of these systems is invertible as $\det G[1:i,1:i] = \operatorname{covol} \Lambda_i^2 > 0$. As such, by Cramer's formulas, $\det G[1:i-1:1:i-1] N_{i,j}$ is an integer and thus $\|\pi_i(v_i)\|_2^2 > \det G[1:i-1,1:i-1]$. This means that the coefficient of $M_{i,j}$ is bounded by $\|v_i\| / \|\pi_j(v_j)\| \leq \sqrt{\det G[1:i-1,1:i-1]} \|v_i\| \leq \|\mathcal{B}\|_{\max}^i$, which is the announced result.

The total cost in term of binary operations is then bounded by $O\left(d^6 B^3\right)$.

Remark (On the differences between the reduction strategies)**.** *As mentioned, two natural strategies can be used to reduce a lattice, the progressive one (like the original* LLL *we presented) and the global one (or* flattening out *strategy). Even though both of them would lead to a reduced basis at the of the process, the overall behavior is quite different and in particular, the way we can analyze the process differs significantly. The analysis of the progressive strategy is* global*: as the reduction is not over, we can assert properties of the current basis. We only have a global information coming from the upper bound on the norm of the profile. Analyzing the flattening out strategy can be done* locally *by looking at the process as a dynamical system which update the whole basis at each iteration, giving a bound on the profile of the basis. Going further in this direction make us design a* parallel variant *of the reduction where we alternate passes of reduction on shifted blocks as in the following scheme:*



This reduction behaves similarly to a well known dynamical process: the profile evolves as a discretized heat diffusion over a 1 dimensional compact[6]*. In particular, the characteristic time of the diffusion process is quadratic in the diameter of the compact. For the reduction algorithm, we can prove that the global number of rounds required to reduce the basis is a $O\left((\operatorname{rk} \Lambda)^2\right)$, which is quadratic in the "diameter" of the basis. A full analysis using this model is performed for more general reductions in Chapter 5.*

6 That is, encoded by the evolution equation $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$.

2.4.6 On the complexity of lattice reduction

We shall now prove that lattice reduction is no easier than linear algebra on a finite field $\mathbf{F}_p$ for a large enough prime p. Let us now define the related complexity problems:

2.4.6.1. *Siegel reduction problem.* We shall define a reduction notion which is slightly weaker than the LLL-reduction notion, called Siegel reduction.

Definition 2.4.2 (Siegel reduction)**.** *A basis $\mathcal{B}$ of a lattice is said to be T-Siegel-**reduced** for $T > 0$ when the following conditions are satisfied:*

$$\forall i < j, \quad \left|\langle v_j, \pi_i(v_i)\rangle\right| \leq \frac{1}{2}\|\pi_i(v_i)\|^2 \quad \textit{(Size-Reduction condition)} \tag{2.5}$$

$$\forall i, \quad T\|\pi_i(v_i)\| \leq \|\pi_{i+1}(v_{i+1})\| \quad \textit{(Siegel condition)} \tag{2.6}$$

The LLL-algorithm produces Siegel-reduced bases with parameter T for any lattice and any $T > \frac{2}{\sqrt{3}}$. The corresponding search problem is written as:

Problem (Siegel reduction for well conditiond bases)**.** *Given an rank d integer lattice Λ given as a matrix A of dimension d with $\|A\|, \|A^{-1}\| \leq 2^B$, find a $\sqrt{2}$-Siegel reduced basis of Λ.*

2.4.6.2. *A linear algebra problem over finite fields.*

Definition 2.4.3 (Kernel problem)**.** *Given a square matrix A of dimension d over $\mathbf{F}_p$, output a matrix K such that $AK = 0$ and the number of columns of K is* $\dim\ker A$.

2.4.6.3. *A reduction for the reduction problem.* The reduction from the Siegel problem to the kernel problem is very straightforward and almost lossless:

Theorem 2.4.2. *If one can solve the Siegel-reduction problem in dimension $2d$ with parameter B, then one can solve the kernel problem in dimension d for any prime $p \leq 2^{B/2-d-1}d^{-1}$ with the same complexity, up to a constant.*

Proof. Let A,p be the input of the kernel problem. The matrix

$$L = \begin{pmatrix} d2^{2d}p^2\,\mathrm{Id}_d & pd2^{2d}A \\ 0 & \mathrm{Id}_d \end{pmatrix}$$

is given to the Siegel reduction oracle. The output is of the form

$$\begin{pmatrix} 0 & * \\ K & * \end{pmatrix}$$

where we maximize the number k of columns of K. The reduction returns this matrix K. We now prove that it is a basis of the kernel of A.

We have $\|L\| \leq 2d^2 2^{2d} p^2 \leq 2^B$ and $\|L^{-1}\| \leq 2d$ which is also less than 2^B since $p \geq 2$. It is clear that vectors in $L\mathbf{Z}^{2d}$ of the form $\begin{pmatrix} 0 \\ x \end{pmatrix}$ are exactly the integer solutions of $Ax = 0 \bmod p$. We let QR be the QR-decomposition of AU. Let K' be a basis of $\ker A$, where entries are integers smaller than p. Then, since U is unimodular, there is an integer matrix V such that

$$AUV = \begin{pmatrix} 0 \\ K' \end{pmatrix}.$$

If V has no nonzero entries $V_{i,j}$ with $i > k$, so that that the output is correct. Hence, we consider v a column of V where it is not the case. First, we have $\|AUv\| \leq \sqrt{d}p$. Second, as Q is orthogonal, we have $\|AUv\| = \|Rv\| \geq R_{i,i}$. Third, the definition of k implies that $R_{k+1,k+1} \geq d2^{2d}p$. As the lattice is reduced and $i > k$, we have $R_{i,i} \geq R_{k+1,k+1}2^{-2d}$. We conclude that:

$$\sqrt{d}p \geq \|AUv\| \geq R_{k+1,k+1}2^{-2d} \geq dp$$

which is incorrect. ■

As we expect the kernel problem to have a complexity of at least an $\Omega(d^\omega B)$, we can expect the same for the Siegel reduction problem, and therefore of the LLL-reduction.

2.4.7 On the behavior of the LLL-reduction and the influence of the δ-relaxation

2.4.7.1. *On the average behavior of LLL* As shown in [95] for Siegel-reduced bases, a reduced basis chosen uniformly at random behaves as the worst-case allowed by the final inequalities. By contrast, bases produced by the LLL algorithm are usually much better than this worst-case. Let us give an example of this phenomena, which already appears in dimension as small as *three*.

Example. *Let $\alpha \in]1, \sqrt{4/3}]$ and define the vectors*

$$\begin{aligned} b_1 &= (\alpha^2 \quad 0 \quad 0) \\ b_2 &= (\alpha\sqrt{\alpha^2 - 1} \quad \alpha \quad 0). \\ b_3 &= (0 \quad \sqrt{\alpha^2 - 1} \quad 1) \end{aligned}$$

We have $\|b_3\| < \|b_2\| \leq \|b_1\|$. Denote by Λ_α the Euclidean lattice spanned by b_1, b_2, b_3. This lattice has only two LLL-reduced bases : $L_3 = [b_1, b_2, b_3]$ and the size-reduction of $R_3 = [b_3, b_2, b_1]$ (we invite the reader to refer to Appendix 2 for a proof of this result). One could intuitively think that the output distribution of LLL on random[7] bases of Λ_α is uniform, or at least close to the uniform distribution. Interestingly, this is not the case: the basis L_α is

7 Even if the natural notion of random lattice is fairly easy to describe, as a normalization of a Haar measure over the moduli space $\mathrm{Gl}(n, \mathbf{R})/\mathrm{Sl}(n, \mathbf{Z})$, it appears that giving a "natural

indeed obtained less than 25% *of the time:* LLL *selects more often the basis with the shortest first vector.*

This example confirms the observation that the LLL procedure "selects" some reduced bases among all the possible LLL-reduced ones, and therefore that in practice the bases output by LLL are *better* (for the norm of the vectors, or the density of the successive sublattices, among other quality measures) than the worst case predictions. Going further in this direction, recall that we have proved that in the worst-case, the δ-LLL-reduction of a lattice of rank d yields a first vector satisfying:

$$\|v\| \leq \left(\frac{1}{\delta - 1/4}\right)^{-\frac{d}{4}} \operatorname{covol}(\Lambda)^{\frac{1}{d}}.$$

However, as remarked in [130], the average output has shorter vectors, in particular, on average, the first vector satisfies:

$$\|v\| \approx 1.02^{-\frac{d}{4}} \operatorname{covol}(\Lambda)^{\frac{1}{d}},$$

for relaxation factor δ close to 1, meaning that LLL favors bases with shorter than expected vectors.

2.4.7.2. *Phase transitions in the δ-LLL-reduction* We have seen that the LLL reduction can be thought as a relaxed and algorithmic version of Hermite's inequality. Of course this relaxation parameter influences the reduction, as it forces the reduction to end in polynomial-time. Hence, we can wonder on the impact of this parameter on global behavior of LLL.

The comportment of the δ-LLL reduction does not seem to be smooth: we can even observe a *phase transition* phenomena. For instance if we go back to our previous example L_α it appears experimentally that the probability of outputting the so-called *dark basis* of the lattice discontinuously change at $\frac{1}{\alpha^2}$. For $\frac{3}{4} \leq \delta < \alpha^{-2}$ the probability is close to one half, whereas as $\delta \geq \alpha^{-2}$, it collapses to 0.25. Figure 4 shows an instance of this phase transition for $\alpha = 1.07$.

Even though the worst-case behavior of LLL is quite easy to state, its average case is still surprising and not-well understood (in particular, even if some models using dynamical system exists [45, 116], there is no rigourous analysis of the LLL-reduction in the average case). Therefore it is invaluable to have access to a fast reduction algorithm having the same execution trace as original LLL to pursue extensive practical studies and experiments. This will be one of our goals when designing a fast and sound reduction process for arbitrary real lattices, in Chapter 4.

definition" of a random basis is not as trivial. Indeed, the naive way of selecting a fixed basis and multiplying by some unimodular transformation is not satisfactory as it introduce significant bias in the behavior of LLL. A possible (and consistent with other methods) solution consists in sampling random vectors according to the discrete Gaussian distribution of sufficiently large variance and transforming such a generating family in a basis with the algorithm of Micciancio and Goldwasser [120]. A complete discussion on this topic would bring us far from the goals of this manuscript, but we point out the preprint [5] for an in-depth treatment of this problem.

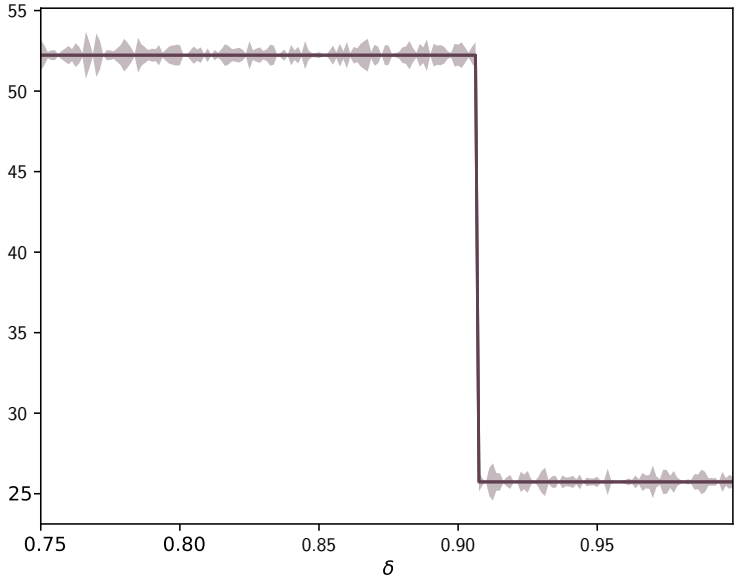


Figure 4: Phase transition on the behavior of δ-LLL for the lattice Λ_α. Probability of outputting $[b_1, b_2, b_3]$ after the δ-LLLreduction of random basis of Λ_α for $\alpha = 1.07$.

2.5 BEYOND THE LLL REDUCTION: REDUCTION WITH SVP ORACLES

2.5.1 Towards Hermite-Korkine-Zolotarev reduction

2.5.1.1. *First minima of the successive projected lattices.* Let $(v_1, \dots, v_d)$ be an LLL-reduced basis. From the definition of the LLL-reduction, we remark that for any index $1 \leq i < d$, the basis $(\pi_i(v_i), \pi_i(v_{i+1}))$ is a Gauss-reduced basis of the lattice $\Lambda_i^* = \pi_i(v_i\mathbf{Z} \oplus v_{i+1}\mathbf{Z})$. From Proposition 2.3.1, it implies that $\pi_i(v_i)$ reaches the first minima of Λ_i^*. A possible generalization of this observation is to require the vector $\pi_i(v_i)$ to reach the first minima of the lattice $\pi_i\left(v_i\mathbf{Z} \oplus \cdots \oplus v_{i+\beta}\mathbf{Z}\right)$ for some parameter $\beta > 2$. This is in substance the definition of the β-Hermite Korkine-Zolotarev reduction notion.

Definition 2.5.1 (β-BKZ reduction). *A basis $\mathcal{B}$ of a lattice is said to be β-BKZ-**reduced** fo5 certain parameters $\beta > 1$, if the following conditions are satisfied:*

$$\forall i < j, \quad \left|\langle v_j, \pi_i(v_i)\rangle\right| \leq \frac{1}{2}\|\pi_i(v_i)\|^2 \quad \textit{(Size-Reduction condition)} \tag{2.7}$$

$$\forall i, \quad \|\pi_i(v_i)\| = \lambda_1\left(\pi_i\left(v_i\mathbf{Z} \oplus \cdots \oplus v_{i+\min(\beta,\mathrm{rk}(\Lambda))}\mathbf{Z}\right)\right) \quad \textit{(β-Minima condition)} \tag{2.8}$$

2.5.1.2. *Pushing β towards infinity, HKZ reduction.* Strengthening the minima condition by letting β grows to infinity yields the Hermite-Korkine-Zolotarev reduction notion: hence the basis $(v_1, \ldots, v_d)$ of a lattice Λ is HKZ reduced if it is weakly-reduced and if the vector $\pi_i(v_i)$ reaches the first minima of the projected lattice $\pi_i(\Lambda)$. This reduction notion is quite natural in the sense that the size-reduction aims at controlling the contribution of the sublattice $(v_1, \ldots, v_{i-1})$ to the vector v_i, whereas the minima condition minimizes the remaining part. Of course in dimension two this notion corresponds exactly to Gauss reduction.

This reduction notion has been originally introduced by Korkine and Zolotarev for quadratic forms in the seminal article [99] and reintroduced by Hermite in his second letter to Jacobi [78].

2.5.2 From slide-reduction to DBKZ

We now turn to the effective computation of a β-HKZ-reduced basis. In all of the following, we suppose that we are given a SVP oracle $\mathcal{O}$, which, when fed with a basis of a lattice Λ, returns a shortest vector of Λ.

2.5.2.1. *A simple algorithm for HKZ reduction.* It is easy to unfold Definition 2.5.1 to construct an algorithm computing an HKZ-reduced basis from the oracle $\mathcal{O}$:

1. Find a shortest vector v_1 of Λ,
2. Complete v_1 in a basis of Λ, and derive a basis of $\pi_1(\Lambda)$ from it by the Gram-Schmidt orthogonalization.
3. Recursively find an HKZ-reduced basis $(v'_2, \ldots, v'_d)$ of $\pi_1(\Lambda)$.
4. Lift each of the v'_i into v_i by adding multiples of v_1, so that the resulting basis $(v_1, \ldots, v_d)$ is size-reduced.

It is clear that the running-time of this algorithm is asymptotically polynomial in the running time of the oracle $\mathcal{O}$. Thanks to the results of Section 2.2, we know that solving exactly these instances of SVP is NP-hard. Practically speaking, the best algorithms known for this problem run in exponential time ($O(2^n)$ [1] for proved algorithms, and $2^{0.292d+o(d)}$ for heuristic sieving algorithms [11]).

As far as the β-HKZ reduction is concerned, the current state of the art of reduction algorithm is divided into two categories: on the one hand the *proved* algorithms, epitomized by the slide reduction of Gama and Nguyen [56] and the other hand, the practical heuristic BKZ variants, culminating in the recent self-dual BKZ of Micciancio and Walter [121].

2.5.2.2. *The Block-Korkine-Zolotarev algorithm.* In [145], Schnorr describes a block generalization of the LLL-algorithm, parametrized by the block-size β which works essentially as follow:

1. Call the LLL reduction to shrink the size of the basis vector $(v_1, \ldots, v_d)$.
2. For each of the block $(v_i, \ldots, v_{\min(i+\beta-1,d)})$
   - Find a shortest vector v of the corresponding projected lattice.
   - If the vector v has a shorter norm than $\pi_i(v_i)$, then construct the lattice Λ' generated by the lift of v and the vectors $v_i, \ldots, v_{\min(i+\beta-1,d)}$ of the block. Reduce this lattice using the LLL variant for generating families to avoid altering the lattice spanned by the block by reducing the size of its first vector.
   - Replace the block by the basis obtained at the previous step.
3. Repeat until a β-HKZ reduced is obtained.

However, no polynomial bound on the number of SVP oracle calls is known for BKZ. In fact, the experiments of [57] suggest that this number of calls does not even grow polynomially!

2.5.2.3. *A proved algorithm for the β-HKZ problem.* In [56], Gama and Nguyen introduced a different block reduction algorithm, called *slide reduction*. It is also parameterized by a block size β, which is required to divide the lattice dimension, but uses a SVP oracle only in dimension β. A basis $\mathcal{B}$ is slide-reduced with block size β, if $\mathcal{B}[1,k]$ is HKZ reduced, $\pi_2(\mathcal{B}[2,k+1])$ is dual-SVP-reduced, and $\pi_{k+1}(\mathcal{B}[k+1,n])$ is slide-reduced. The slide-reduction process starts by alternately SVP-reducing all blocks $\pi_{ik+1}(\mathcal{B}[ik+1,(i+1)k])$ and running LLL on $\mathcal{B}$. Once no more changes occur, the blocks $\pi_{ik+2}(\mathcal{B}[ik+2,(i+1)k+1])$ are dual-SVP reduced. This entire process is iterated until no more changes occur. Although enjoying a clean analysis, the slide-reduction algorithm has been reported by its author to be sensibly inferior to BKZ in experiments.

2.5.2.4. *The heuristic BKZ algorithm.* Instead of relying on proved algorithms, the *heuristic* variants of BKZ algorithm are used in practice since they provide very good experimental running time and better-than-expected output quality.

The search for a shortest vector is usually carried by enumeration techniques with computations carried out in floating-point representation, coupled with so-called *pruning* strategies (see [6, 7] for instance) to provide early cuts in the enumeration tree and sensibly reduce the running time.

However, the *caveat* of the variants of BKZ algorithm is their notorious difficulty to analyze properly. Actually, while there is no polynomial bound on the number of calls BKZ makes to the SVP oracle, Hanrot, Pujol, and Stehlé showed in [75] that one can *abort* a variant of BKZ reduction after a polynomial number of calls to the oracle and still provably achieve reasonable bounds on the size of the reduced vectors. Up to the work of Micciancio and Walter [121] the *statu-quo* was that a complex simulation procedure seemed to be the way to predict the result of their application to an instance (see [33] for an instance of this simulation strategy).

2.5.2.5. *The DBKZ algorithm.* By providing a back-and-forth strategy coupled with enumeration in the dual lattice, the *dual block Korkine-Zolotarev* algorithm provides an algorithm with practical performances comparable to the BKZ variants but with provable and easy bounds on its running time. We provide the blueprint of this reduction in Algorithm 12.

Algorithm 12 — DBKZ reduction

Input : $(v_1, \dots, v_d)$ a basis of a rank d lattice Λ, a SVP oracle $\mathcal{O}$
Output : A k-reduced basis of Λ

do
  for $i = 1$ **to** $d - \beta + 1$ **do**
    Find v such that $\|\pi_i(v)\| = \lambda_1\left(\pi_i(v_i), \dots, \pi_i(v_{i+\beta-1})\right)$
    $v_i, \dots, v_{i+\beta-1} \leftarrow \mathbf{LLL}(v, v_i, \dots, v_{i+\beta-1})$
  end for
  for $i = n - \beta + 1$ **downto** 1 **do**
    $d_i, \dots d_{i+\beta-1} \leftarrow \left(\pi_i(v_i), \dots, \pi_i(v_{i+\beta-1})\right)^\vee$
    Find v such that $\|\pi_i(v)\| = \lambda_1\left(d_i, \dots, d_{i+\beta-1}\right)$
    $v_i, \dots, v_{i+\beta-1} \leftarrow \mathbf{LLL}(v, v_i, \dots, v_{i+\beta-1})$
  end for
while *progress is done*
return $(v_1, \dots, v_d)$

The *while progress is done* condition is not completely straightforward to implement as it can stuck the algorithm in infinite loops, as mentioned in [121]. However, Micciancio and Walter show by a dynamical system analysis it is possible to abort the execution of the algorithm after a polynomial number of rounds and still achieve provable bounds on the output. This technique offers a tradeoff between the time spent and the quality of the output by looking at blocks of size $\beta \leq n$, as stated in the following theorem. For the purposes of this manuscript it is encompassed in the following theorem, which can be easily derived from the original bound of [121] and from the results on practical enumeration from [11].

Theorem 2.5.1. *There exists an algorithm ouputting a vector v of a lattice Λ satisfying:*

$$\|v\| \leq \beta^{\frac{n-1}{2(\beta-1)}} \cdot (\operatorname{covol}\Lambda)^{\frac{1}{n}}.$$

Such a bound can be achieved in time $\mathrm{Poly}(n, \log\|B_0\|)\left(\frac{3}{2}\right)^{\beta/2+o(\beta)}$, where B_0 is the input basis.

Proof. The bound we get is a direct consequence of [121, Theorem 1]. We only replaced the *Hermite constant* γ_β by an upper bound in $O(\beta)$ as presented in Paragraph 2.1.4.2. The cost analysis is derived from a study of [121, Algorithm 1] (with the abort technique after a polynomial number of rounds), and the complexity of the *Shortest Vector Problem* (SVP) is below $\left(\frac{3}{2}\right)^{\beta/2+o(\beta)}$ operations, according to [11]. ∎

Remark (Complexity bound for polynomial-time γ-SVP)**.** *This last result ensures that solving γ-SVP for approximation factors in $O\left(2^{\frac{n \log\log n}{\log n}}\right)$ is possible in polynomial time, by instantiating the DBKZ algorithm with block-size $\beta = \Theta(\log n)$.*

We conclude this chapter by exposing a somewhat folklore result on the reduction of lattices with small determinants. This result is crucial for the design of the algorithm of Chapter 6.

2.6 ON THE REDUCTION OF LATTICES WITH SMALL DETERMINANTS

Let us examine the right-hand side of the equation of Theorem 2.5.1: $(\operatorname{covol}\Lambda)^{\frac{1}{d}} \cdot \beta^{\frac{2d}{\beta}}$. Both the d-th root of the determinant and exponential factor in d appear in this term. Generically, the determinant part prevails. However, when the determinant is small, the second term can be of the same order of magnitude than the root of the determinant. In this case one could desire to balance the contributions of these two expressions. A natural idea is then to reduce a lattice of smaller dimension in order to reduce this approximation factor. We fix the block-size $\beta \leq d$ and look at the output of DBKZ performed on the sublattice Λ' generated by the m first vectors $b_1, \ldots, b_m$ of an HNF basis. From Theorem 2.5.1, we have

$$\|v\| \leq \beta^{\frac{m}{2\beta}} \cdot (\operatorname{covol}\Lambda')^{\frac{1}{m}} \leq \beta^{\frac{m}{2\beta}} \cdot (\operatorname{covol}\Lambda)^{\frac{1}{m}}.$$

The condition we require on the determinant of the lattice is $\operatorname{covol}\Lambda \leq \beta^{\frac{n^2}{2\beta}}$: otherwise, for every $m \leq d$, the term $(\operatorname{covol}\Lambda)^{\frac{1}{m}}$ is dominating. Assuming $\operatorname{covol}\Lambda \leq \beta^{\frac{n^2}{2\beta}}$, we identify the optimal sub-dimension m in $\{\beta, \ldots, n\}$ depending on β that minimizes this upper bound: it corresponds to the balance between the two factors, that is $m = \left\lfloor \sqrt{2\beta \log_\beta(\operatorname{covol}\Lambda)} \right\rceil$. We fix m to this value and we obtain the following corollary.

Corollary 2.6.1. *For any integer lattice Λ of rank d such that $\operatorname{covol}\Lambda \leq \beta^{\frac{n^2}{2\beta}}$, using DBKZ reduction with block-size β on the first $m = \left\lfloor \sqrt{2\beta \log_\beta(\operatorname{covol}\Lambda)} \right\rceil$ vectors permits to output a short vector v that satisfies*

$$\log\|v\| \leq (1 + o(1)) \sqrt{\frac{2\log\beta}{\beta} \deg\Lambda}.$$

This algorithm runs in time $\mathrm{Poly}(n, B) \cdot \left(\frac{3}{2}\right)^{\beta/2 + o(\beta)}$, with B a bound on the bitsize of the coefficients of the input basis.

Proof. We consider the sublattice of dimension m, for m as defined above. The condition on the determinant of Λ ensures that our value of m is effectively lower than d. Then, by Theorem 2.5.1 and Lemma 2.2.2, we have

$$\|v\| \leq \beta^{\frac{m}{2\beta}} \cdot (\operatorname{covol}\Lambda)^{\frac{1}{m}} = \beta^{\sqrt{(2/\beta)\log_\beta(\operatorname{covol}\Lambda)}\left(1 + o(1)\right)},$$

which yields the announced result — the $(1 + o(1))$ factor appears because of the integer approximation of m. ■

Remark. *Thanks to Corollary 2.6.1, we want to point out that choosing block-size $\beta = \log(\operatorname{covol}\Lambda)^{\frac{1}{3}}$ when it is smaller than d allows to describe an algorithm that runs in time $\mathrm{Poly}(n, \log\|B_0\|) \cdot \left(\frac{3}{2}\right)^{\beta/2+o(\beta)}$ and outputs a vector of norm less than $\beta^{\sqrt{2}\beta\left(1+o(1)\right)}$. This remark will be the crux of the descent technique of Chapter 6.*

3

ON LATTICES OVER ORDERS IN NUMBER FIELDS

We now dive into the generalization of lattices to the higher dimensional number theoretical setting.

This chapter introduces and exposes the elementary properties of an algebraic generalization of Euclidean lattices. Recall that a lattice is a free $\mathbf{Z}-$module endowed with an Euclidean metric. Since $\mathbf{Z}$ is the ring of integers of the field $\mathbf{Q}$, a natural question arising from this sole definition is what we obtain under a field extension, that is an $\mathscr{O}_{\mathbf{K}}-$module endowed with a suitable metric, i.e. compatible with the algebraic structure of the underlying number field.

Remark. *These lattices defined over rings of integers appear in Arakelov theory as (Hermitian) vector bundles on the arithmetic curve Spec$\mathscr{O}_{\mathbf{K}}$. But they were already considered by Humbert in 1939 [88], in the equivalent language of Hermitian forms rather than lattices, and have been further studied in the spirit of classical lattice theory under the name* Humbert forms. *However, it is remarkable, as noticed by Yves André in [4], that these two trends of research on the* same *object ignored each other for more than 50 years.*

3.1 RELATIVE STRUCTURE OF MODULES OVER TOWERS

3.1.1 Projectiveness of ring of integers

In Theorem 1.2.2, we have proved that the ring of integers of a number field is free as $\mathbf{Z}$-module. It is a module not just over $\mathbf{Z}$, but also over any intermediate ring of integers. That is to say, if we have a tower of number fields $\mathbf{Q} \subseteq \mathbf{K} \subseteq \mathbf{L}$, then the ring $\mathscr{O}_{\mathbf{L}}$ can be viewed as an $\mathscr{O}_{\mathbf{K}}$-module. Since $\mathscr{O}_{\mathbf{L}}$ is finitely generated over $\mathbf{Z}$, it is also finitely generated over $\mathscr{O}_{\mathbf{K}}$, as being the same module seen under scalar extension. However, even though $\mathscr{O}_{\mathbf{L}}$ is free over $\mathbf{Z}$, it is not necessarily free over $\mathscr{O}_{\mathbf{K}}$.

Example (Inspired by Keith Konrad's [98])**.** *Let* $\mathbf{K} = \mathbf{Q}(i\sqrt{6})$ *and* $\mathbf{L} = \mathbf{K}(i\sqrt{3})$. *Then,*

$$\mathscr{O}_{\mathbf{L}} = \frac{1+i\sqrt{3}}{2}\mathscr{O}_{\mathbf{K}} \oplus \frac{1}{i\sqrt{3}}\mathfrak{a},$$

for $\mathfrak{a}$ *being generated by* $(3, \sqrt{-6})$, *but* $\mathscr{O}_{\mathbf{L}}$ *is not free over* $\mathscr{O}_{\mathbf{K}}$.

Proof. Suppose that $\mathscr{O}_{\mathbf{L}}$ is free over $\mathscr{O}_{\mathbf{K}}$.

- Then, we would have:

$$\mathscr{O}_{\mathbf{L}} = \frac{1+i\sqrt{3}}{2}\mathscr{O}_{\mathbf{K}} \oplus \mathscr{O}_{\mathbf{K}}.$$

Indeed, take $E = (e_1, e_2)$ an $\mathscr{O}_{\mathbf{K}}$-basis of $\mathscr{O}_{\mathbf{L}}$. We claim that since $F = \left(1, \frac{1+i\sqrt{3}}{2}\right)$ is a $\mathbf{K}$ basis of $\mathbf{L}$, the change of basis matrix $M = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{pmatrix}$ transforms the basis E to the basis F had determinant in $\mathscr{O}_{\mathbf{K}}^{\times}$. Indeed, $\mathbf{L}/\mathbf{K}$ is Galois of degree 2, with the non-trivial automorphism given by the conjugation $\sigma(i\sqrt{3}) = -i\sqrt{3}$. Then, M transforms $(\sigma(e_1), \sigma(e_2))$ into $(1, \frac{1-i\sqrt{3}}{2})$. Therefore:

$$\begin{pmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{pmatrix} \begin{pmatrix} e_1 & \sigma(e_1) \\ e_2 & \sigma(e_2) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \frac{1+i\sqrt{3}}{2} & \frac{1-i\sqrt{3}}{2} \end{pmatrix},$$

and so by taking the determinant of both sides of this equality we find:

$$\det(M)(e_1\sigma(e_2) - e_2\sigma(e_1)) = -i\sqrt{3}.$$

As such,

$$N_{\mathbf{K}/\mathbf{Q}}(\det(M))^2 N_{\mathbf{K}/\mathbf{Q}}\left((e_1\sigma(e_2) - e_2\sigma(e_1))^2\right) = 9.$$

This last equation asserts that $N_{\mathbf{K}/\mathbf{Q}}(\det(M))$ is either 1 or 3. Remark that for any $a, b \in \mathbf{Z}$, $N_{\mathbf{K}/\mathbf{Q}}\left(a + i\sqrt{6}b\right) = a^2 + 6b^2$, which can not be equal to 3. Then $N_{\mathbf{K}/\mathbf{Q}}(\det(M)) = 1$, as claimed.

- We now prove that F can not be a basis of $\mathscr{O}_{\mathbf{L}}$. Suppose it is the case, then, since $\sqrt{2} = \frac{i\sqrt{6}}{i\sqrt{3}} \in \mathscr{O}_{\mathbf{L}}$ we can write, $\sqrt{2} = \alpha + \beta\frac{1+i\sqrt{3}}{2}$ for some $\alpha, \beta \in \mathscr{O}_{\mathbf{K}}$. Then by the action of the Galois element σ we find: $-\sqrt{2} = \alpha + \beta\frac{1-i\sqrt{3}}{2}$. But then if we take the difference of these two last equations we would have $2\sqrt{2} = i\beta\sqrt{3}$, implying $4 \times 2 = -9\beta^2$ by squaring, but remark that $-8/9 \notin \mathscr{O}_{\mathbf{K}}$, which is a contradiction.

■

As such, $\mathscr{O}_{\mathbf{L}}$ might not be free over $\mathscr{O}_{\mathbf{K}}$ but it is at least projective, as defined in the definition given in Section 1.1.4:

Proposition 3.1.1. *Let $\mathbf{Q} \subseteq \mathbf{K} \subseteq \mathbf{L}$ be tower of number fields, then $\mathscr{O}_{\mathbf{L}}$ is a projective module over $\mathscr{O}_{\mathbf{K}}$.*

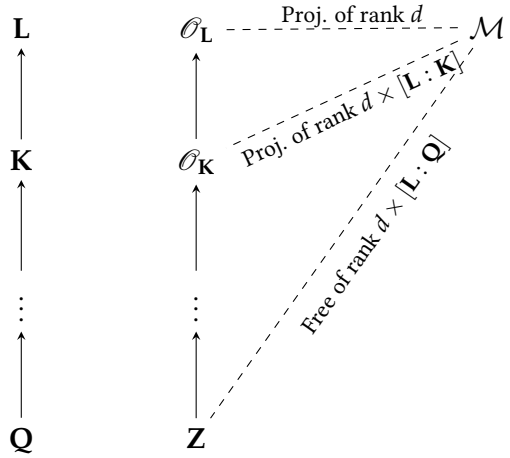
Proof. $\mathscr{O}_{\mathbf{L}}$ is clearly finitely generated over $\mathscr{O}_{\mathbf{K}}$ and has no torsion. Let $X = (x_1, \ldots, x_n)$ be a generating set for $\mathscr{O}_{\mathbf{L}}$ over $\mathscr{O}_{\mathbf{K}}$. Let $T = (t_1, \ldots, t_k)$ be a set of $\mathscr{O}_{\mathbf{K}}$ linearly independent elements of $\mathscr{O}_{\mathbf{L}}$. Then $k \leq n$. Suppose that T is maximal for the cardinality, so that, $t_1\mathscr{O}_{\mathbf{K}} + \cdots + t_k\mathscr{O}_{\mathbf{K}}$ is free of rank k. Without loss of generality—up to composing by a suitable isomorphism—we can suppose that t_i is the i-th vector of the standard basis of the free module

$\mathcal{O}_{\mathbf{K}}^k$. For any element $x \in \mathcal{O}_{\mathbf{L}}$, the maximality of T implies that there exists an integer $\alpha_x \in \mathcal{O}_{\mathbf{K}} \setminus \{0\}$ such that $\alpha_x \cdot x \in t_1\mathcal{O}_{\mathbf{K}} + \cdots + t_k\mathcal{O}_{\mathbf{K}} = \mathcal{O}_{\mathbf{K}}^k$. Doing so on each of the spanning set X yields a family $\alpha_1, \ldots, \alpha_n$, verifying that $\alpha_i x_i \in \mathcal{O}_{\mathbf{K}}^k$ for each i. Hence the product of these elements also satisfies:

$$\alpha_1 \cdots \alpha_n \mathcal{O}_{\mathbf{L}} \subseteq \mathcal{O}_{\mathbf{K}}^k.$$

Since the α_i are non-zero, $\alpha = \alpha_1 \cdots \alpha_n$ is also non-zero and, as such, is invertible in $\mathbf{K}$, so that as an $\mathcal{O}_{\mathbf{K}}$ module, $\mathcal{O}_{\mathbf{L}} \cong \alpha\mathcal{O}_{\mathbf{L}}$. Hence, it is a direct factor of the free module $\mathcal{O}_{\mathbf{K}}^k$. ∎

3.1.1.1. *Relative structure of projective modules* Let $\mathbf{Q} \subseteq \mathbf{K} \subseteq \mathbf{L}$ be a tower of number fields and $\mathcal{M}$ a projective module over the ring $\mathcal{O}_{\mathbf{L}}$. Then $\mathcal{M}$ is also projective over $\mathcal{O}_{\mathbf{K}}$. Indeed, it is isomorphic to a direct sum of fractional ideals of $\mathcal{O}_{\mathbf{L}}$. Each of them is itself a projective module over $\mathcal{O}_{\mathbf{K}}$. Its rank over $\mathcal{O}_{\mathbf{K}}$ of course satisfies $\mathrm{rk}_{\mathcal{O}_{\mathbf{K}}}(\mathcal{M}) = \mathrm{rk}_{\mathcal{O}_{\mathbf{L}}}(\mathcal{M})[\mathbf{L} : \mathbf{K}]$. Hence $\mathcal{M}$ can be seen as a module over any intermediate ring of integers between $\mathcal{O}_{\mathbf{L}}$ and $\mathbf{Z}$, as depicted as follows:



3.2 NUMBER FIELDS AND CANONICAL EUCLIDEAN STRUCTURE

3.2.1 Archimedean embeddings, canonical norm

Let $\mathbf{L}$ be a number field of degree n and $\mathcal{O}_{\mathbf{L}}$ its maximal order. Denote by α a primitive element of $\mathbf{L}$. Then, as seen in Chapter 1, there are exactly n distinct embeddings, ie. field homomorphisms, of $\mathbf{L}$ into $\mathbf{C}$. We define the i-th embedding $\sigma_i : \mathbf{L} \to \mathbf{C}$ as the morphism mapping α to α_i. We distinguish embeddings induced by real roots, the *real embeddings*, from embeddings coming from complex roots, the *complex embeddings*, and sort them as:

- $\sigma_1, \ldots, \sigma_r$ for the real ones,
- $\sigma_{r+1}, \ldots, \sigma_n$ for the complex ones, paired so that $\sigma_{n+i} = \overline{\sigma_{n+i}}$.

The $\mathbf{L}$-vector space $\mathbf{L_R} = \mathbf{L} \otimes_{\mathbf{Q}} \mathbf{R}$ is naturally a $\mathbf{R}-$algebra and any of the embeddings σ_i defines a morphism from $\mathbf{L_R}$ to $\mathbf{C}$, as $\sigma_i(\alpha \otimes r) = r\sigma_i(\alpha)$, so that we have an isomorphism of algebra, called the *Archimedean embedding*:

$$\Sigma : \left| \begin{array}{ccc} \mathbf{L_R} & \longrightarrow & \mathbf{R}^r \times \mathbf{C}^c \\ x & \longmapsto & (\sigma_1(x), \dots, \sigma_r(x), \sigma_{r+1}(x), \dots \sigma_n(x)) \end{array} \right. .$$

This embedding allows to define a symmetric bilinear form on $\mathbf{L_R}$, which is positive definite and endows $\mathbf{L_R}$ with a natural Hermitian structure given by

$$\langle a, b \rangle_\sigma = \sum_{i=1}^{n} \overline{\sigma_i(a)} \sigma_i(b).$$

The corresponding norm is called the *canonical norm*, or T_2 norm as defined in [12]. We can make the canonical embedding an isometry, by endowing the space $\mathbf{R}^r \times \mathbf{C}^c$ with the product:

$$\langle a, b \rangle_\sigma = \sum_{i=1}^{r} a_i b_i + 2 \sum_{i=r+1}^{c} \Re\left(a_i \overline{b_i}\right).$$

3.2.2 Extension to vector spaces.

Let d be a non-negative integer. We can construct the real vector space $\bigoplus_{i=1}^{d} \mathbf{L_R} = \mathbf{L_R^d}$. It is obviously a space of dimension dn, but also a free $\mathbf{L_R}$-module of rank n. Since each of the summand is an Euclidean space with the canonical norm, we can endow this space with a natural Euclidean structure by setting:

$$\langle x, y \rangle = \sum_{i=1}^{d} \langle x_i, y_i \rangle_\sigma,$$

for any $x = (x_1, \dots, x_d) \in \mathbf{L_R^d}$ and $y = (y_1, \dots, y_d) \in \mathbf{L_R^d}$.

Example. *Let us consider the most simple case, where $\mathbf{L} = \mathbf{Q}$, that is, a number field of absolute degree 1. Then by definition of the tensor product $\mathbf{L} \otimes_{\mathbf{R}} \mathbf{R} \cong \mathbf{R}$. There is a unique embedding into $\mathbf{C}$ (mapping 1 on 1, that is the identity map), and as such the canonical norm is simply $\|x\| = \sqrt{x\bar{x}} = |x|$, the absolute value. Thus, when taking $\mathbf{L_R^d} \cong \mathbf{R}^d$ the previous construction yields the product: $\langle x, y \rangle = \sum_{i=1}^{d} x_i y_i$, which is nothing more than the usual scalar product in the canonical basis of $\mathbf{R}^d$.*

Remark. *By linearity, this definition is equivalent to set:*

$$\langle x, y \rangle = \sum_{\sigma : \mathbf{L} \to \mathbf{C}} \overline{\sigma(x)}^T \sigma(y)$$

when extending the embeddings $\sigma : \mathbf{L} \to \mathbf{C}$ pointwisely over vectors: $\sigma(x) = (\sigma(x_1), \dots, \sigma(x_d))$.

Now that we have defined a natural Euclidean metric on this space, we can twist it to get all the metrics compatible with the underlying algebraic

structure. Recall that in the case where $\mathbf{L} = \mathbf{Q}$, once the standard scalar product is given, one can define the notion of self-adjoint operators, each of them defining a new Euclidean metric on the space. We can emulate this construction in this more general case.

3.2.3 (Additive) Humbert forms

There have been a certain number of attempts to develop a notion generalization of quadratic forms in high degree. In the 1940s, Paul Humbert presented two papers ([88] and [89]) about the reduction theory of quadratic forms over number fields. It appears that his notion is not strictly speaking a form but a tuple of quadratic forms, which we will call Humbert forms in the sequel, to follow the term coined by Icaza in [90].

Remark. *We only describe here the notion of* additive *Humbert forms (as opposed to* multiplicative *Humbert forms), as this are the ones we will get interested in, from the reduction point of view. We let the interested reader refer to [40] for a precise account on multiplicative Humbert forms and their relation to higher degree Voronoï theory.*

3.2.3.1. *Humbert forms by twisting the embeddings* Recall that a positive Hermitian (resp. Symmetric) quadratic form A of rank d is said to be positive definite if for all non-zero $x \in \mathbf{C}^d$ (resp. $x \in \mathbf{R}^d$), we have $\langle Ax, x\rangle > 0$, where $\langle \cdot, \cdot \rangle$ is the canonical inner product of $\mathbf{C}^d$ (resp. $\mathbf{R}^d$).

To fix the notations of this section, let n, d be two non-negative integers. We take a number field $\mathbf{L}$ of degree n, with r real embeddings and c complex ones. Set V to be the space $\mathbf{L}_{\mathbf{R}}^d$.

Definition 3.2.1. *A Humbert form $\mathcal{A}$ of rank d over $\mathbf{L}$ is a d-tuple $(A_\sigma)_{\sigma:\mathbf{L}\to\mathbf{C}}$ consisting of r positive definite symmetric forms of rank d and s positive definite Hermitian forms of rank d. The* evaluation *of $\mathcal{A}$ on vectors $x, y \in \mathbf{L}_{\mathbf{R}}^d$ is:*

$$\mathcal{A}(x, y) = \sum_{\sigma:\mathbf{L}\to\mathbf{C}} A_\sigma(\sigma(x), \sigma(y)).$$

Therefore, through the structure isomorphism given by the Archimedean embedding giving $\mathbf{L}_{\mathbf{R}}^d \cong (\mathbf{R}^d)^r \times (\mathbf{C}^d)^c$, a Humbert form can be seen as a *quadratic twist* of the canonical extension of the T_2 inner product at each of the embeddings.

3.2.3.2. *Matrix representation* Let us fix a basis of $\mathbf{L}_{\mathbf{R}}^d$ and thus of each of the image $\sigma(\mathbf{L}_{\mathbf{R}}^d)$ at each embedding. We fix a Humbert form $\mathcal{A} = (A_\sigma)_{\sigma:\mathbf{L}\to\mathbf{C}}$. By the choice of the basis, each of the quadratic form A_σ can be encoded as a matrix, which is Hermitian positive definite for the complex embeddings and Symmetric positive definite for the real ones. If we also denote by $(A_\sigma)_{\sigma:\mathbf{L}\to\mathbf{C}}$ this matrix representation, we have for any $x, y \in \mathbf{L}_{\mathbf{R}}^d$:

$$\mathcal{A}(x, y) = \sum_{\sigma:\mathbf{L}\to\mathbf{C}} \overline{\sigma(x)}^T A_\sigma \sigma(y) \tag{3.1}$$

so that the canonical norm of $\mathbf{L}$ is exactly the evaluation of the Humbert form $(\mathrm{Id}_1)_{\sigma:\mathbf{L}\to\mathbf{C}}$, and its extension to $\mathbf{L}^d$ is the form $(\mathrm{Id}_d)_{\sigma:\mathbf{L}\to\mathbf{C}}$.

Given a Humbert form $\mathcal{A} = (A_\sigma)_{\sigma:\mathbf{L}\to\mathbf{C}}$ given as a tuple of complex matrices, remark that for any $1 \leq i, j \leq d$, the n-tuple $s = ((A_\sigma)_{i,j})_{\sigma:\mathbf{L}\to\mathbf{C}}$ defines uniquely an element of $\mathbf{L}_\mathbf{R}$, given by $\Sigma^{-1}(s)$ for Σ being the Archimedean embedding. Hence, $\mathcal{A}$ can be encoded as a matrix $A \in \mathbf{L}_\mathbf{R}^{d\times d}$ and as such we have by Equation 3.1:

$$\mathcal{A}(x, y) = \langle x, Ay \rangle.$$

This latest expression recovers the usual construction of the evaluation of a Hermitian or Symmetric quadratic form as a matrix.

Remark. *We can go further in this analogy and characterize the matrices encoding Humbert forms, as the cone of matrices satisfying:*

$$\forall x, y \in \mathbf{L}_\mathbf{R}^d, \langle x, Ay \rangle = \langle Ax, y \rangle$$

and $\langle x, Ax \rangle > 0$ *for any non-zero* $x \in \mathbf{L}_\mathbf{R}^d$. *This is equivalent to requiring that* $A = A^\dagger$, *where* $\cdot \mapsto \cdot^\dagger$ *is the composition of the transposition with the unique involution lifting the complex conjugation through* Σ^{-1}. *We let the reader refer to the thesis of Camus [30] for a more complete account on the relation between Humbert forms and the structure of the space End*$(\mathbf{L}_\mathbf{R}^d)$.

3.3 LATTICES OVER NUMBER FIELDS

We now generalize the notion of Euclidean lattice to the higher-degree context. Recall from Section 2.1 that a lattice is a finitely generated free $\mathbf{Z}$-module Λ endowed with a Euclidean structure on its real ambient space $\Lambda \otimes_\mathbf{Z} \mathbf{R}$. To extend this definition we want to replace the base-ring $\mathbf{Z}$ by the ring of integers $\mathcal{O}_\mathbf{L}$ of a number field $\mathbf{L}$. However, the freeness condition is actually a bit too strong. Indeed we have seen that in a tower of field $\mathbf{Q} \subseteq \mathbf{K} \subseteq \mathbf{L}$, the module $\mathcal{O}_\mathbf{L}$ seen over the Dedekind domain $\mathcal{O}_\mathbf{L}$ is not necessarily free. Hence, if we use as definition of an $\mathcal{O}_\mathbf{L}$-lattice to be a free $\mathcal{O}_\mathbf{L}$-module, then such lattices would not necessarily be lattices over $\mathcal{O}_\mathbf{K}$. Relaxing the freeness into projectiveness is however sufficient as $\mathcal{O}_\mathbf{L}$ is always a projective $\mathcal{O}_\mathbf{K}$-module.

Definition 3.3.1 ($\mathcal{O}_\mathbf{L}$-lattice)**.** *Let* $\mathbf{L}$ *be a number field. An* $\mathcal{O}_\mathbf{L}$*-lattice—or algebraic lattice over* $\mathcal{O}_\mathbf{L}$*—is a projective* $\mathcal{O}_\mathbf{L}$*-module* Λ *of finite rank, endowed with a Humbert form on the ambient vector space* $\Lambda \otimes_{\mathcal{O}_\mathbf{L}} \mathbf{R}$.

Remark. *We can rewrite Definition 3.3.1, as being a projective* $\mathcal{O}_\mathbf{L}$*-module endowed with a family of Hermitian forms* $\|\cdot\|_\sigma$ *on the spaces* $\Lambda \otimes_{\mathcal{O}_\mathbf{L}} \mathbf{C}$, *which are invariant by conjugation. This definition corresponds to the notion of* Hermitian vector bundles over the arithmetic curve *Spec*$\mathcal{O}_\mathbf{L}$, *as studied in Arakelov geometry. For a very detailed account on this topic, we invite the reader to refer to the monograph of Bost [22].*

3.3.1 Relative structure

3.3.1.1. *Direct Image of an algebraic lattice.* Let us consider $\mathbf{Q} \subset \mathbf{K} \subset \mathbf{L}$ a tower of number fields, and denote by $\mathbf{Z} \subset \mathscr{O}_\mathbf{K} \subset \mathscr{O}_\mathbf{L}$ their respective ring of integers. Let Λ be a lattice over the top ring $\mathscr{O}_\mathbf{L}$. We have seen in Section 3.1 that as a module, the underlying module of Λ can be viewed as an $\mathscr{O}_\mathbf{K}$ module. We can go further by *descending* the whole Euclidean structure to make it compatible with the algebraic structure on $\mathscr{O}_\mathbf{L}$. More precisely for an $\mathscr{O}_\mathbf{L}$-lattice $(\Lambda, (A_\sigma)_\sigma)$ we define its *direct image* over $\mathscr{O}_\mathbf{K}$ as being $(\Lambda, (A'_\sigma)_{\sigma'})$ where for each embedding $\sigma' : \mathbf{K} \to \mathbf{L}$ we have for any $x, y \in \Lambda$:

$$A_{\sigma'}(x, y) = \sum_{\sigma : \mathbf{L} \to \mathbf{C}} A_\sigma(x, y),$$

where the sum is taken over the $[\mathbf{L} : \mathbf{K}]$ embeddings $\sigma : \mathbf{L} \to \mathbf{C}$ such that the restriction of σ to $\mathbf{K}$ is σ'.

Remark. *When viewing an algebraic lattice as a Hermitian vector bundle over an arithmetic curve, this descent corresponds exactly to performing a* Weil restriction of scalar *(or* Weil descent*).*

3.3.1.2. *Direct Image to a Euclidean lattice.* In particular when descending all the way through the base of the tower $\mathbf{Z}$, we can define the direct image Λ_* of Λ over $\mathbf{Z}$, where Λ is seen as a projective—and therefore free— module over $\mathbf{Z}$ with the Euclidean structure being given by:

$$\|x\|^2 = \sum_{\sigma : \mathbf{L} \to \mathbf{C}} A_\sigma(x, x),$$

for any $x \in \Lambda$, the sum being taken over all the $[\mathbf{L} : \mathbf{Q}]$ embeddings.

3.3.1.3. *An important example: ideal lattices.* Let $\mathbf{L}$ be a number field, and $\mathfrak{a}$ an ideal of $\mathscr{O}_\mathbf{L}$. Then $\mathfrak{a}$ is by definition a projective module over the Dedekind domain $\mathscr{O}_\mathbf{L}$. Equipped with the canonical norm described in Section 3.2, it is a $\mathscr{O}_\mathbf{L}$-lattice. Its rank over $\mathscr{O}_\mathbf{L}$ is of course 1, as

$$\mathfrak{a} \otimes_{\mathscr{O}_\mathbf{L}} \mathbf{L}_\mathbf{R} \cong \mathbf{L}_\mathbf{R},$$

and its direct image $\mathfrak{a}_*$ is an Euclidean lattice of rank $[\mathbf{L} : \mathbf{Q}]$. It is called the *ideal lattice* attached to $\mathfrak{a}$.

Ideal lattices have become ubiquitous in the context of modern cryptography with the rise of so-called *lattice based cryptography*. We let the reader refer to Chapter 7, for a more detailed account on the use of such lattices.

Example (An ideal over a small cyclotomic)**.** *Let $\mathbf{L} = \mathbf{Q}[\zeta_8]$ the cyclotomic field of conductor 8 and thus of degree 4 over $\mathbf{Q}$. For notational simplicity, from now on, we denote $\zeta = \zeta_8$. The Archimedean embeddings of $\mathbf{L}$ to $\mathbf{C}$ are given by the $\sigma_i : \zeta \mapsto \zeta^i$ for $i = 1, 3, 5, 7$. Let us take the ideal $\mathfrak{a}$ generated by the element $g = 1 - \zeta$. Then we have:*

$$N(\mathfrak{a}) = N_{\mathbf{L}/\mathbf{Q}}(g) = 2.$$

Since $\mathscr{O}_{\mathbf{L}} = \mathbf{Z}[\zeta]$, a simple choice for a $\mathbf{Z}$-basis of $\mathfrak{a}$ is given by:

$$\begin{aligned}\mathfrak{a} &= (1-\zeta)\left(\mathbf{Z} \oplus \zeta\mathbf{Z} \oplus \zeta^2\mathbf{Z} \oplus \zeta^3\mathbf{Z}\right) \\ &= (1-\zeta)\mathbf{Z} \oplus (\zeta-\zeta^2)\mathbf{Z} \oplus (\zeta^2-\zeta^3)\mathbf{Z} \oplus (\zeta^3+\zeta)\mathbf{Z}.\end{aligned}$$

The corresponding Euclidean structure is then given by the Gram matrix:

$$\mathcal{A} = \begin{pmatrix} 8 & -4 & 0 & 4 \\ -4 & 8 & -4 & 0 \\ 0 & -4 & 8 & -4 \\ 4 & 0 & -4 & 8 \end{pmatrix}$$

by applying the descent formulas of previous section. For instance the canonical norm of the element $(1-\zeta)$ is

$$\|1-\zeta\| = \left\|\left(1-\zeta, 1-\zeta^3, 1-\zeta^5, 1-\zeta^7\right)^T\right\|_2^{\frac{1}{2}} = 2\sqrt{2}.$$

3.3.2 Generalized invariants of algebraic lattices

The geometric invariants attached to a Euclidean lattice are of course generalizable to the algebraic setting. In all of the following we consider an $\mathscr{O}_{\mathbf{L}}$-lattice (Λ, A).

3.3.2.1. *Covolume.* We can extend the definition of covolume of the lattice.

We could give an algebraic definition to the covolume of an algebraic lattice by noticing that the volume of the fundamental cell $\mathbf{R}^n/\Lambda$ was obtained by taking the determinant of the Gram-matrix of any basis of Λ. We can do the same in the algebraic case, by taking this time the algebraic norm of this determinant —which is $\mathscr{O}_{\mathbf{L}}$-valued. However on the contrary to $\mathbf{Z}$ modules, an algebraic lattice can not be solely described by a basis, but instead by a pseudo-basis. Let us give an intuition of the impact of the phenomena on the covolume. To do so, consider a basis $v_1, \dots, v_d$ in $\mathbf{L}^d$ and $\mathfrak{a}$ an ideal of $\mathscr{O}_{\mathbf{L}}$. Construct two modules $\mathcal{M} = v_1\mathscr{O}_{\mathbf{L}} \oplus \cdots \oplus v_d\mathscr{O}_{\mathbf{L}}$ and $\mathcal{M}' = v_1\mathscr{O}_{\mathbf{L}} \oplus \cdots \oplus v_{d-1}\mathscr{O}_{\mathbf{L}} \oplus v_d\mathfrak{a}$. Since $\mathfrak{a}$ is contained in $\mathscr{O}_{\mathbf{L}}$, we have $\mathcal{M}' \subset \mathcal{M}$. Hence $\mathcal{M}$ is denser than $\mathcal{M}'$, in the sense that it contains "more points". As such, the covolume of $\mathcal{M}$ should be smaller than the covolume of $\mathcal{M}'$. Intuitively, $\mathcal{M}$ contains $\ell = \left|\mathcal{M}/\mathcal{M}'\right|$ more points than $\mathcal{M}'$, as $\mathcal{M}$ decomposes in a disjoint union of ℓ copies of $\mathcal{M}'$ indexed by the split of the exact sequence $0 \to \mathcal{M}' \to \mathcal{M} \to \mathcal{M}/\mathcal{M}' \to 0$. Moreover, remark that the quotient $\mathcal{M}/\mathcal{M}'$ has cardinality $\left|\mathscr{O}_{\mathbf{L}}/\mathfrak{a}\right|$, that is $N(\mathfrak{a})$. Hence the covolume of $\mathcal{M}'$ should be $N(\mathfrak{a})$ larger than the covolume of $\mathcal{M}$, yielding $\operatorname{covol}\mathcal{M}' = N(\mathfrak{a})\operatorname{covol}\mathcal{M} = N(\mathfrak{a})N_{\mathbf{L}/\mathbf{Q}}\left(\det\left(b_i^\dagger b_j\right)\right)$. The same reasoning extends to arbitrary projective modules, since we can represent it in the same way as $\mathcal{M}'$ with $\mathfrak{a}$ being its Steinitz class (see Theorem 1.1.2 for

the definition of this class). All in all we get the following definition (linearly twisted by A, to accommodate the fact that we can use an arbitrary Euclidean metric instead of the canonical one):

Definition 3.3.2. *Let* $\mathbf{L}$ *be a number field and* (Λ, A) *be an* $\mathscr{O}_{\mathbf{L}}$*-lattice of rank* d*, described by a pseudo-basis* $((v_1, \mathfrak{a}_1), \dots, (v_d, \mathfrak{a}_d))$*. Its* covolume *is defined by*

$$\operatorname{covol}\Lambda = N(\mathfrak{a}_1 \cdots \mathfrak{a}_d)\sqrt{N_{\mathbf{L}/\mathbf{Q}}(\det[\mathcal{G}])},$$

where $\mathcal{G}$ *is the Gram matrix of* $v_1, \dots, v_d$*, that is* $\mathcal{G} = V^\dagger A V$ *for* $\dagger$ *the involution presented in Paragraph 3.2.3.1. Its* degree *is defined as the logarithm of its covolume.*

It is linked to the covolume of the direct image over $\mathbf{Z}$ as follows:

$$\operatorname{covol}(\Lambda_*) = \operatorname{covol}(\Lambda)|\Delta_{\mathbf{L}}|^{\frac{1}{2}},$$

by definition of the absolute discriminant given in Section 1.3.3. In particular we have: For any $\mathfrak{a}$ of $\mathscr{O}_{\mathbf{L}}$, then

$$\operatorname{covol}(\mathfrak{a}_*) = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})|\Delta_{\mathbf{L}}|^{\frac{1}{2}}.$$

Example (An ideal in a small cyclotomic)**.** *Let us return to the case of the ideal* $\mathfrak{a}$ *generated by* $(1-\zeta)$ *in the cyclotomic field* $\mathbf{L}$ *of conductor* 8*. A pseudo-basis of this ideal is* $((1-\zeta), \mathscr{O}_{\mathbf{L}})$*, as being a principal ideal. Thus its covolume for the canonical norm is*

$$\operatorname{covol}\mathfrak{a} = N(\mathscr{O}_{\mathbf{L}})\sqrt{N_{\mathbf{L}/\mathbf{Q}}(\zeta^\dagger\zeta)} = 1 \times \sqrt{N_{\mathbf{L}/\mathbf{Q}}(\zeta^3 - \zeta + 2)} = 2.$$

Remark that this result is consistant with the fact that another choice for the pseudo-basis of this lattice which is $(1, \mathfrak{a})$*, which would give:*

$$\operatorname{covol}\mathfrak{a} = N(\mathfrak{a})\sqrt{N_{\mathbf{L}/\mathbf{Q}}(1^\dagger 1)} = N(\mathfrak{a}) \times \sqrt{1} = 2.$$

We have seen that the direct image of this lattice over $\mathbf{Z}$ *is isometric to the lattice* $(\mathbf{Z}^n, \mathcal{A})$*. Hence its covolume is* $\operatorname{covol}\mathfrak{a}_* = \sqrt{\det\mathcal{A}} = 32$*. Since the discriminant* Δ *of* $\mathbf{L}$ *is 256, we easily check that* $\operatorname{covol}(\mathfrak{a}_*) = 32 = 2 \times \sqrt{256} = N_{\mathbf{L}/\mathbf{Q}}(\mathfrak{a})\Delta_{\mathbf{L}}^{\frac{1}{2}}$.

3.3.2.2. *Minima of the lattice* The minima are defined in the same way that for usual Euclidean lattice, that is:

$$\lambda_1(\Lambda) = \min_{v\in\Lambda, x\neq 0}\|v\|_\Lambda$$

and by taking the direct image of an $\mathscr{O}_{\mathbf{L}}$-lattice, we can bound the first minima by the correct normalization of its covolume:

Remark. *In particular, we can define a generalization of the Hermite constant for algebraic lattices and thereof for Humbert forms. It should nonetheless be noticed that this constant depends on the Steinitz class of the modules (or the so-called $\mathfrak{A}$-invariant in the work of [30]):*

$$\gamma_{d,\mathfrak{a}} = \max_{\Lambda} \left[\frac{\lambda_1(\Lambda)}{(\operatorname{covol}\Lambda)^{\frac{1}{d}}} \right],$$

where the maximum is taken over $\mathscr{O}_{\mathbf{L}}$-lattices sharing the same Steinitz class $\mathfrak{a}$. For a precise treatment of this question in a more general context (which encompasses Thunder's constant, Rankin's contant, and adelic version of them) we invite the reader to refer to the work of Watanabe and Coulangeon [41].

3.4 ON THE REDUCTION THEORY FOR ALGEBRAIC LATTICES

As for Euclidean lattices, we want to approximate as efficiently as possible short vectors of algebraic lattices. Therefore, it is natural to wonder if a generalization of the algorithms presented in Chapter 2 can be provided to tackle this problem. Of course, we can always descend an algebraic lattice to $\mathbf{Z}$ by taking its direct image and reduce the corresponding lattice. But doing so, we are multiplying its rank by the absolute degree of the field where it is defined! The work of Fieker and Stehlé [54] gives a precise analysis of the LLL reduction algorithm for algebraic lattices, using this technique.

Hence even small rank lattices can become very difficult to computationally reduce if the absolute degree of the field is too high. In particular, this is also the case for ideal lattices over large degree cyclotomic fields, which are now ubiquitous in cryptography[1]. In 2016, Chris Peikert, in a survey on ideal lattice cryptography actually asks the following question [133]: *For worst-case problems on ideal lattices, especially in cyclotomic rings, are there algorithms that substantially outperform the known ones for general lattices?* Being able to efficiently reduce algebraic lattices is then a matter of interest, not only in computational number theory, but also for cryptanalytical purposes.

3.4.1 Generalizing the LLL algorithm

In the following let us fix a number field $\mathbf{L}$ of absolute degree n and an $\mathscr{O}_{\mathbf{L}}$-lattice (Λ, A) of rank d. Without loss of generality and to ease the presentation of algorithms, we can suppose that the operator A is the identity operator. Let us suppose that Λ is given by a pseudo-basis $(v_i, \mathfrak{a}_i)_{1 \leq i \leq d}$ where $v_i \in \mathbf{L}^d$ and $\mathfrak{a}_i$ are fractional ideals of $\mathbf{L}$.

As the LLL algorithm is a basic building block of all reduction procedures, it is natural to wonder if an algebraic generalization is possible, which would not directly reduce a lattice by descending it over $\mathbf{Z}$. A first step in this

1 We give a longer account on the cryptographical perspectives of algebraic lattices in the third part of this manuscript.

direction comes from the work of Fieker and Pohst in [53], which aims at reducing pseudo-bases. The blueprint of this reduction is given as pseudo-code in Algorithm 13. We voluntarily let the meaning of what is a *reduced*-pseudo-basis unclear for the moment and discuss it afterward. The reduction mimics the LLL algorithm but it replaces the canonical norm over $\mathbf{R}^d$ by the canonical norm of $\mathbf{L}^d$.

Algorithm 13 — Prototype of reduction of [53]

Input : $((v_1, \mathfrak{a}_1), \dots, (v_d, \mathfrak{a}_d))$ a pseudo-basis of a rank d lattice Λ
Output : A "reduced" basis of Λ

1 $k \leftarrow 2$
2 **while** $k \leq d$ **do**
3 $\quad (v_1^*, \dots, v_d^*) \leftarrow$ **Orthogonalize**$(v_1, \dots, v_d)$
4 $\quad$ **for** $j = k-1$ **downto** 1 **do**
5 $\quad\quad$ Find $x \in \mathfrak{a}_{k-1}\mathfrak{a}_k^{-1}$ which minimizes $\left\| x - \frac{{v_{k-1}^*}^\dagger v_k}{{v_{k-1}^*}^\dagger v_{k-1}^*} \right\|_{\mathbf{L}/\mathbf{Q}}$
6 $\quad\quad$ $v_k \leftarrow v_k - x v_{k-1}$
7 $\quad$ **end for**
8 $\quad$ **if** $\delta \|{v_{k-1}}^*\|_{\mathbf{L}/\mathbf{Q}} \leq \left\| (v_k^*) + \frac{{v_{k-1}^*}^\dagger v_k}{{v_{k-1}^*}^\dagger v_{k-1}^*} \cdot v_{k-1}^* \right\|_{\mathbf{L}/\mathbf{Q}}$ **then**
9 $\quad\quad$ $k \leftarrow k+1$
10 $\quad$ **else**
11 $\quad\quad$ **Swap**(v_{k-1}, v_k)
12 $\quad\quad$ **Swap**$(\mathfrak{a}_{k-1}, \mathfrak{a}_k)$
13 $\quad\quad$ $k \leftarrow \max(k-1, 2)$
14 **end while**
15 **return** $((v_1, \mathfrak{a}_1), \dots, (v_d, \mathfrak{a}_d))$

The orthogonalization process is then performed in the exact same way as in the rational case: denoting the orthogonalized family obtained from $(v_i)_{1 \leq i \leq d}$ by v_i^* as a shorthand for the orthogonal projection over the $\mathbf{L}$-vector space spanned by the vectors $v_1, \dots, v_{i-1}$. The computation is done inductively as follows: $v_1^* = v_1$ and for all $1 < i \leq d$, and

$$v_i^* = v_i - \sum_{j=1}^{i-1} \frac{{v_i}^\dagger v_j^*}{{v_j^*}^\dagger v_j^*} v_j,$$

the vector-vector product being here understood as a duality bracket. This algorithm can be modified to reduce according to various notions of reduction, depending on whether we choose to consider the canonical norm or the algebraic norm. Indeed, line 8 of Algorithm 13 corresponds to the so-called Lovàsz condition in the original LLL algorithm and should be adapted to our

algebraic context. However, as mentioned by the authors, it can be modified in multiple ways. For instance, we can lift it using either the algebraic norm:

$$\forall i, \quad \delta N_{\mathbf{L}/\mathbf{Q}}({v_{k-1}}^*) \leq N_{\mathbf{L}/\mathbf{Q}}\left((v_k^*) + \frac{{v_{k-1}^*}^\dagger v_k}{{v_{k-1}^*}^\dagger v_{k-1}^*} \cdot v_{k-1}^*\right),$$

or twist it with the norm of the ideals of the pseudo-basis so that it corresponds to the condition stated on covolumes in Section 2.4.2:

$$\forall i, \quad \delta N(\mathfrak{a}_i) N_{\mathbf{L}/\mathbf{Q}}({v_{k-1}}^*) \leq N(\mathfrak{a}_{i-1}) N_{\mathbf{L}/\mathbf{Q}}\left((v_k^*) + \frac{{v_{k-1}^*}^\dagger v_k}{{v_{k-1}^*}^\dagger v_{k-1}^*} \cdot v_{k-1}^*\right).$$

The size-reduction itself can also be made to deal with algebraic norms, if looking for an element x such that $N_{\mathbf{L}/\mathbf{Q}}\left(x - \frac{{v_{k-1}^*}^\dagger v_k}{{v_{k-1}^*}^\dagger v_{k-1}^*}\right)$ is minimal.

Using the same arguments as for the traditional LLL algorithm, one can prove that the generalized reduction terminates. It should, however, be noticed that the size-reduction step requires to find exactly a closest vector in $\mathscr{O}_{\mathbf{L}}$. Indeed, over $\mathbf{Z}$, this step aims at finding the closest lattice vector in a coset of the form $x + \mathbf{Z}v$. In the algebraic case, this translates into seeking for the closest vector in a coset of the shape $x + \mathscr{O}_{\mathbf{L}}v$. Up-to-our knowledge, no exponential speedup on calling a CVP oracle on the corresponding Euclidean lattice $(\mathscr{O}_{\mathbf{L}})_*$ exists to solve such instances. Even in the case where we are able to find such an element x, we can not get any bound on the size of the vectors output by the reduction process. To find such an estimate, as in the rational case, one would need that

$$\left\| x - \frac{{v_{k-1}^*}^\dagger v_k}{{v_{k-1}^*}^\dagger v_{k-1}^*} \right\|_{\mathbf{L}/\mathbf{Q}} < \delta < 1,$$

or

$$N_{\mathbf{L}/\mathbf{Q}}\left(x - \frac{{v_{k-1}^*}^\dagger v_k}{{v_{k-1}^*}^\dagger v_{k-1}^*}\right) < \delta < 1.$$

This is not the case in general, even for quadratic fields, as soon as the discriminant is too large. However, for fields satisfying such a condition, we can adapt quite straightforwardly the proof of the LLL reduction. It appears that this condition implies that the field is norm-Euclidean.

3.4.2 Reduction over norm-Euclidean rings

3.4.2.1. *On norm-Euclidean domains.*

Definition 3.4.1. *Let R be an integral domain. A Euclidean function, or stathsme, on R is a function $f : R\{0\} \to \mathbf{N} \setminus \{0\}$, satisfying the division-with-remainder property: if a and $b \neq 0$ are in R, then there exists q and r in R such that:*

$$a = bq + r \quad \text{and either} \quad r = 0 \text{ or } f(r) < f(b).$$

Let $\mathbf{L}$ be a number field and $\mathscr{O}_{\mathbf{L}}$ be its ring of integers. Then $\mathscr{O}_{\mathbf{L}}$ is said to be *norm-Euclidean* if it is Euclidean for the algebraic norm function $N_{\mathbf{L}/\mathbf{Q}}$.

Example. *It is noticeable that imaginary quadratic rings of integers are not necessarily norm-Euclidean, even when being principal:*

- $\mathbb{Q}(\sqrt{-19})$ *is principal and not Euclidean.*
- $\mathbb{Q}(\sqrt{69})$ *is Euclidean and not norm-Euclidean*

Lenstra proved that small cyclotomics are norm-Euclidean:

Theorem 3.4.1 (Lenstra [111])**.** *Let f be a non negative integer so that $\phi(f) < 16$ and $f \notin \{16, 24\}$. Then $\mathscr{O}_{\mathbf{Q}(\zeta_f)} \cong \mathbf{Z}[\zeta_f]$ is norm-Euclidean.*

3.4.3 Unimodularity of Euclidean rings

Lemma 3.4.1. *Let $\mathbf{L}$ be a number field whose ring of integers $\mathscr{O}_{\mathbf{L}}$ is norm-Euclidean, then:*

$$m_{\mathbf{L}} = \sup_{x \in \mathbf{L}} \inf_{y \in \mathscr{O}_{\mathbf{L}}} |N_{\mathbf{L}/\mathbf{Q}}(x - y)|.$$

Proof. Let $\mathbf{L}$ be a number field such that $m_{\mathbf{L}} > 1$. By definition,

$$m_{\mathbf{L}} < \inf\{\varepsilon > 0 \,|\, \forall x \in \mathbf{L}\, \exists \gamma \in \mathscr{O}_{\mathbf{L}} |N_{\mathbf{L}/\mathbf{Q}}(x - \gamma)| < \varepsilon\}$$

. Then, there exists $\alpha, \beta \in \mathscr{O}_{\mathbf{L}}$ such that for any $x \in \mathbf{L}$:

$$|N_{\mathbf{L}/\mathbf{Q}}(\beta - \alpha x)| > |N_{\mathbf{L}/\mathbf{Q}}(\beta)|$$

contradicting the definition of norm-Euclideanity. ∎

From this theorem one can easily find out which imaginary quadratic fields are norm-Euclidean:

Lemma 3.4.2. *Let $d > 0$ be a squarefree non-negative integer. Then the imaginary quadaratic field $\mathbf{Q}[i\sqrt{d}]$ is norm-Euclidean iff $d \in \{1, 2, 3, 7, 11\}$.*

Proof. Direct, by computing the Euclidean cell

$$\sup_{x \in \mathbf{L}} \inf_{y \in \mathscr{O}_{\mathbf{L}}} |N_{\mathbf{L}/\mathbf{Q}}(x - y)|.$$

∎

3.4.3.1. *Principality of norm-Euclidean domains* The arithmetic of ideals in a norm-Euclidean domain is very easy, indeed, every ideal is principal:

Lemma 3.4.3. *Let R be an Euclidean domain, then any ideal $\mathfrak{a}$ is principal.*

Proof. Let $\mathfrak{a}$ be an ideal of R. Then consider a non-zero element $x \in \mathfrak{a}$ for which the stathsme f is minimal, which always exists since f is valued in $\mathbf{N} \setminus \{0\}$. Suppose that $\mathfrak{a}$ is not generated by x. Then there exists $y \in \mathfrak{a} \setminus xR$. But then there exist $q, r \in R$ so that $y = xq + r$, with $f(r) < f(x)$ or $r = 0$. By hypothesis on y, r can not be 0 and it belongs to $\mathfrak{a}$ as being equal to $y - xq$, contradicting the minimality of $f(x)$. ∎

Therefore, any projective module over a Euclidean ring is free, and a pseudo-basis is nothing more than a basis in the usual sense.

3.4.4 Generalizing the LLL reduction to the norm-Euclidean case

In the case where the field $\mathbf{L}$ is norm-Euclidean, we can perform the reduction with the algebraic norm, that is by adaptating the reduction process of Fieker and Pohst. The corresponding code is given in Algorithm 14.

Algorithm 14 — Norm Euclidean Reduction

Input : $(v_1, \dots, v_d)$ a basis of a rank d $\mathcal{O}_{\mathbf{L}}$-lattice Λ.
Output : A "reduced" basis of Λ.

1 $k \leftarrow 2$
2 **while** $k \leq d$ **do**
3 $\quad (v_1^*, \dots, v_d^*) \leftarrow$ **Orthogonalize**$(v_1, \dots, v_d)$
4 $\quad$ **for** $j = k-1$ **downto** 1 **do**
5 $\quad\quad$ Find $x \in \mathcal{O}_{\mathbf{L}}$ which minimizes $N_{\mathbf{L}/\mathbf{Q}}\left(x - \frac{{v_{k-1}^*}^{\dagger} v_k}{{v_{k-1}^*}^{\dagger} v_{k-1}^*}\right)$
6 $\quad\quad$ $v_k \leftarrow v_k - x v_{k-1}$
7 $\quad$ **end for**
8 $\quad$ **if** $\delta N_{\mathbf{L}/\mathbf{Q}}({v_{k-1}}^*) \leq N_{\mathbf{L}/\mathbf{Q}}\left(v_k^* + \frac{{v_{k-1}^*}^{\dagger} v_k}{{v_{k-1}^*}^{\dagger} v_{k-1}^*} \cdot v_{k-1}^*\right)$ **then**
9 $\quad\quad$ $k \leftarrow k+1$
10 $\quad$ **else**
11 $\quad\quad$ **Swap**(v_{k-1}, v_k)
12 $\quad\quad$ $k \leftarrow \max(k-1, 2)$
13 **end while**
14 **return** $(v_1, \dots, v_d)$

This is the algorithm given by Napias in [126]. We can adapt the results presented in Theorem 2.4.1.

Lemma 3.4.4. *Let $\mathbf{L}$ a number field such that $\mathcal{O}_{\mathbf{L}}$ is norm-Euclidean and Λ be an $\mathcal{O}_{\mathbf{L}}$-lattice of rank d. Then, for any basis $(v_1, \dots, v_d)$, reduced by Algorithm 14 with parameter $m_{\mathbf{L}} < \delta < 1$, we have:*

$$\operatorname{covol}(v_1\mathcal{O}_{\mathbf{L}} \oplus \cdots \oplus v_i\mathcal{O}_{\mathbf{L}}) \leq \left(\frac{1}{\delta - m_{\mathbf{L}}}\right)^{\frac{k(d-k)}{2}} \operatorname{covol}\Lambda^{\frac{k}{d}}.$$

This algorithm has been enhanced by Camus in his PhD thesis [30], using similar techniques as [131]. The corresponding complexity for a quadratic number field is a $\mathrm{O}\left(d^4 \log\left(\frac{m_{\mathbf{L}} \prod_{i=1}^{d-1} \gamma_{\mathcal{O}_{\mathbf{L}},r}^{-\frac{r}{d^2}}}{B}\right)\right)$, with B a bound on the bitsize of the coefficients of the imput basis.

The next part of this manuscript aims at providing sound and practical techniques for the reduction of general algebraic lattices, going further than the reductions above mentioned for norm-Euclidean lattices.

Part II

ALGORITHMIC REDUCTION OF ALGEBRAIC LATTICES

This second part aims at developing algorithmic methods to perform the reduction of algebraic lattices. On the one hand we are interested in designing sound and certifiable reductions techniques. On the other hand we also desire to improve the efficiency of the reduction to tackle larger and larger examples. Eventually we demonstrate an application of such reduction algorithms to solve a classical problem in effective number theory, the principal ideal problem.

4

CERTIFIED LATTICE REDUCTION

4.1 LATTICE REDUCTION, CERTIFICATION AND INTERVAL ARITHMETIC

4.1.1 On the approximation of algebraic lattices

Let $\mathcal{L} = (\Lambda, A)$ be an algebraic lattice over the ring of integers $\mathscr{O}_{\mathbf{L}}$ of a number field $\mathbf{L}$. By definition of algebraic lattices, to practically compute the inner product of two vectors $u, v \in \Lambda$, we need to be able to compute the embeddings $\mathbf{L} \to \mathbf{C}$. Hence, this requires to compute the roots of the defining polynomial of $\mathbf{L}$, meaning that in practice that we look for a sufficiently precise approximation of them. Suppose that we want to effectively reduce $\mathcal{L}$. By the discussion conducted in the previous chapter, we can take its direct image $\mathcal{L}_* = (\Lambda_*, A')$ over $\mathbf{Z}$ and reduce it[1]. However, the corresponding inner product A' is *a priori* real-valued and its computation is subjected to the approximation we made when computing the embeddings.

Up-to-our knowledge, no satisfactory estimation[2] of the precision required on this inner product to reduce ideals, and thereof algebraic lattices, appears in the literature. Some authors, like Belabas [12], suggest using some arbitrary approximation and let the LLL reduction operate. However in this setting, the outputted basis has no reason to be LLL-reduced, as mentioned for instance by Cohen in [34]. In some cases, this aspect is not an issue since the reduction was only used to shrink the size of coefficients involved in some computations, but one can't assert any bounds on the norm of elements appearing in these somewhat reduced-bases.

But in the cases where we need to ensure that the output basis is reduced, we need to be able to *certify* the reduction. This implies to control the propagation of approximations during the reduction. A possible strategy to algorithmically tackle this issue is to use *Interval Arithmetic*.

1 More precisely, we would then need to lift the reducted basis to a pseudo-basis of the original lattice $\mathcal{L}$. This computation can be done in polynomial time by adapting the techniques of [54].

2 Buchmann gives in [26] a bound on the required precision to achieve this goal by using a direct approximation of the input basis. However, this bound is computed in terms of a quantity called the *defect* that can be very large and also involves the first minimum of the lattice. Thus it can not be computed independently of the geometry of the input lattice.

4.1.2 On Interval arithmetic

Interval Arithmetic is a representation of reals by intervals—whose endpoints are floating-point numbers—that contain them. Arithmetic operations, in particular the basic operations $+, -, \times, \div$ can be redefined in this context. The main interest of this representation lies in its *certification* property: if real numbers are represented by intervals, the interval resulting from the evaluation of an algebraic expression contains the exact value of the evaluated expression.

For some authors, Interval Arithmetic was introduced by R. Moore in 1962 in his Ph.D. thesis [125]. For others, it can be dated back to 1958, in an article of T. Sunaga [158] which describes an algebraic interpretation of the lattice of real intervals, or even sooner in 1931 as a proposal in the Ph.D. thesis [165] of R.C. Young at Cambridge. Its main asset—calculating directly on sets—is nowadays used to deterministically determine the global extrema of a continuous function [138] or localizing the zeroes of a function and (dis)proving their existence [91]. Another application of Interval Arithmetic is to be able to detect lack of precision at run-time of numerical algorithms, thanks to the guarantees it provides on computations. This can, in particular, be used to design adaptive-precision numerical algorithms.

We aim at transforming and generalizing the LLL algorithm into an adaptive-precision version, so that it can reduce *arbitrary* real lattices—and in particular, the direct image over $\mathbf{Z}$ of algebraic lattices—and is forced to follow a certified flow of execution. More precisely, it uses Interval Arithmetic to validate the size-reduction and exchange steps that occur within the LLL reduction process.

4.1.3 Practical reduction and behavior of LLL

Current implementations of LLL often work with low precision approximations of the Gram-Schmidt vectors in order to greatly speed-up the computations. Indeed, the algorithm works surprisingly well even with such reduced precision, even if some care needs to be taken to avoid infinite loops. Moreover, once the result is obtained, it can be verified efficiently as shown in [162].

The certified algorithms presented in this chapter allow an alternative strategy where we not only certify that the end-result is a reduced basis but also that the algorithm followed a valid computation path to reach it. This strongly deviates from other approaches that have been taken to obtain guaranteed lattice reduced basis. At first, this may seems irrelevant. After all, one might claim that a basis satisfying the end conditions of LLL is what is desired and that the computation path doesn't matter. However we have seen in Chapter 2, that the LLL-reduction tends to select the best bases among the set of reduced bases and presents very peculiar behavior, such as the phase transition effect described in Paragraph 2.4.7.2.

This argues in favor of trying to follow the algorithm definition exactly to better understand the phenomenon. In particular, this option might be invaluable for experiments performed toward analyzing this quality gap and explore the peculiarities of LLL behavior.

4.2 BACK ON LATTICE REDUCTION: FLOATING POINT REPRESENTATION AND PRECISION

To begin with, we go back to the algorithmic reduction of Euclidean lattices and give a short review of the methods used to greatly improve the efficiency of the LLL algorithm by using floating-point representation of the internal values.

4.2.1 Floating point representation

The total cost of the LLL algorithm is dominated by the computation to handle arithmetic on rational values. A first idea of De Weger [164] to overcome this issue is to avoid the use of denominators by multiplying all the quantities by their common denominator. This is slightly more efficient in practice but doesn't improve the asymptotics. Another idea is to remark that the norms of the rational values remain small and to try to use approximations instead of exact values. However, directly replacing rationals in the LLL algorithm by floating-point approximations leads to severe drawbacks. The algorithm might not even terminate, and the output basis is not guaranteed to be LLL-reduced.

The first *provable* floating-point version of the algorithm is due to Schnorr in [146], with complexity $O\big(d^4 \log(B) M(d+B)\big)$, for B being a bound on the bitsize of the coefficients of the input lattice and d its rank. One of the key ingredients to achieve this reduction is to slightly relax the definition of the size-reduction, in order to compensate for the approximation errors introduced by the use of floating-point arithmetic. We call *admissible* any parameters (δ, η) satisfying $1/4 < \delta < 1$, and $1/2 < \eta < \sqrt{\delta}$ and define:

Definition 4.2.1 ((δ,η)-LLL reduction)**.** *Let (δ, η) be admissible parameters. A basis $\mathcal{B}$ of a lattice is said to be (δ,η)-**LLL-reduced** if the following condition is satisfied:*

$$\forall i < j, \quad \big|\langle v_j, \pi_i(v_i)\rangle\big| \leq \eta \|\pi_i(v_i)\|^2 \quad \textit{(Approximate size-reduction condition)} \tag{4.1}$$

together with the Lovász condition, which is kept unchanged from Definition 2.4.1.

Using naive multiplication, the cost of Schnorr's algorithm is cubic in B. The introduction of approximate size reduction removes the need to know with extreme precision values close to half-integers. Instead, approximate size reduction of such values can be achieved by rounding either up or down in an arbitrary (possibly randomized) manner. In our pseudo-code, we use a function called η-Closest-Integer to achieve this rounding, returning an integer at distance at most η of the function's argument.

4.2.2 The L^2 algorithm

The L^2 algorithm is a variant of Schnorr-Euchner version [147] of LLL. By contrast with the original algorithm, L^2 computes the GSO coefficients on the fly as they are needed instead of doing a full orthogonalization at the start. It also uses a lazy size reduction inspired by the Cholesky factorization algorithm. These optimizations yield an improved lattice reduction with running time $O\big(d^5 B(d+B)\big)$.

As usual in lattice reduction, while performing the Gram-Schmidt orthogonalization of $\mathcal{B}$, we also compute QR-decomposition of B into $B^* \cdot M$ where B^* is the matrix representing the $(\pi_i(v_i))_{1\leq i\leq d}$, and M is the upper unitriangular matrix, whose coefficients with $j \geq i$ are $M_{i,j} = \frac{\langle v_j, \pi_i(v_i)\rangle}{\|\pi_i(v_i)\|^2}$. Thus, the Gram matrix associated to the basis, i.e., $G = B^T B$ satisfies:

$$G = M^T \cdot B^{*T} \cdot B^* \cdot M = M^T \cdot D \cdot M$$

where D is a diagonal matrix whose entries are $\|\pi_i(v_i)\|^2$. We denote by R the matrix $D \cdot M$, and thus have $G = R^T \cdot M = M^T \cdot R$.

We give the pseudo-code of the Lazy Size-Reduction procedure as Algorithm 15 and of the L^2 algorithm as Algorithm 16. Both use classical formulas relating R, M and B^* to perform the computations.

Algorithm 15 — η-LazyRed

Input : Initial basis $\mathcal{B} = (v_1, \ldots, v_d)$, with G, R and M.
An integer $1 \leq k \leq d$.
Output : Size-reduces v_k, updates G, R, M and returns $s^{(k)}$

done $\leftarrow$ **false**
while *done* $=$ **false do**
  for $j = 1$ **to** $k-1$ **do**
    $R_{k,j} \leftarrow G_{k,j}$; **for** $i = 1$ **to** $j-1$ **do** $R_{k,j} \leftarrow R_{k,j} - M_{j,i} R_{k,i}$
    $M_{k,j} \leftarrow R_{k,j} / R_{j,j}$;
  end for
  $s_1^{(k)} \leftarrow G_{k,k}$; **for** $j = 2$ **to** k **do** $s_j^{(k)} \leftarrow s_{j-1}^{(k)} - M_{k,j-1} \cdot R_{k,j-1}$
  $R_{k,k} \leftarrow s_k^{(k)}$
  if $(\max_{j<k} |M_{k,j}|) \leq \eta$ **then** done $\leftarrow$ **true**
  else
    for $i = k-1$ **downto** 1 **do**
      $X_i \leftarrow \eta$-**Closest-Integer**$(M_{k,i})$
      for $j = 1$ **to** $i-1$ **do** $M_{k,j} \leftarrow M_{k,j} - X_i M_{i,j}$
    end for
    $v_k \leftarrow v_k - \sum_{i=1}^{k} X_i v_i$; Update G accordingly
  end if
end while

Algorithm 16 — L^2

Parameter : $\delta \in (1/4, 1), \eta \in (1/2, \sqrt{\delta})$.
Input : Initial basis $\mathcal{B} = (v_1, \ldots, v_d)$
Result : A (δ, η)-LLL-reduced basis

Compute $G = G(v_1, \ldots, v_d)$ in exact integer arithmetic
$R_{1,1} \leftarrow G_{1,1}$
$k \leftarrow 2$
while $k \leq d$ **do**
  Apply size reduction η-LazyRed(k)
  $k' \leftarrow k$
  while ($k \geq 2$ **and** $\delta R_{k-1,k-1} > s_{k-1}^{k'}$) **do** $k \leftarrow k - 1$
  $R_{k,k} \leftarrow s_k^{k'}$
  if $k \neq k'$ **then**
    for $i = 1$ **to** $k - 1$ **do** $M_{k,i} \leftarrow M_{k',i}$; $R_{k,i} \leftarrow R_{k',i}$
    $R_{k,k} \leftarrow s_k^{k'}$
    Insert $v_{k'}$ at position k (before v_k) and update matrix G accordingly
  end if
  $k \leftarrow k + 1$
end while
return $(v_1, \cdots, v_d)$

4.2.2.1. *Precision required.* The precision required by the L^2-Algorithm is

$$d \log\left(\frac{(1+\eta)^2}{(\delta-\eta)^2} + \epsilon\right) + o(d)$$

bits for any $\epsilon > 0$, i.e., almost linear in the dimension of the lattice. Moreover, as discussed in [130], it appears that—even though this bound can be shown to be sharp by specific examples—experiments indicate that the number of bits required *on average* is, in fact, lower.

This phenomenom is well-known and is often used in existing algorithms and softwares in the form of a compute-and-verify paradigm. For example, this is default strategy of the well-known FPLLL [159]. It relies on the fact that verifying that a lattice basis is indeed reduced is much less costly than the reduction itself, as shown in [162]. In addition, it is necessary to take several conservative measures in order to prevent the implementation to enter potentially infinite loops.

The approach we propose deviates from this paradigm. Instead of guaranteeing the end-result, we want to make sure that the whole computation follows the mathematical definition of the algorithm. With low-precision approximations, it is unclear how this could be done. However, interval-arithmetic offers a neat solution to achieve this goal.

4.3 INTERVAL ARITHMETIC AND ITS CERTIFICATION PROPERTY

Interval arithmetic is a representation of reals by intervals that contain them. For instance, one can specify a value x with an error ϵ by giving an interval of length ϵ containing x. For example, the constant π can be represented with an error of 10^{-2} by the interval $[3.14, 3.15]$. Interval arithmetic is crucial in the context of *certified* numerical computations, where reals can only be represented with finite precision. For more details, the interested reader can consult an extensive reference, such as [124].

In the following, we denote by $\underline{x}$ a closed interval $[\underline{x}^-, \underline{x}^+]$. We define its *diameter* as the positive real $\underline{x}^+ - \underline{x}^-$ and its *center* as the real $\frac{1}{2}(\underline{x}^+ + \underline{x}^-)$.

Given a real-valued function $f(x_1, \ldots, x_n)$ an interval-arithmetic realization of f is an interval-valued function F such that the interval $F(\underline{x_1}, \ldots, \underline{x_n})$ contains all the values $f(x_1, \ldots, x_n)$ for $(x_1, \ldots, x_n)$ in $\underline{x_1} \times \cdots \times \underline{x_n}$.

If F always returns the smallest possible interval, it is called a *tight* realization, otherwise it is called *loose*. In practice, tight realizations can only be achieved in very simple specific cases. However, even a loose realization is sufficient to certify the correctness of a computation.

Another important property of interval arithmetic is that it can be used to compare numbers in a certified way, as long as the intervals that represent them are disjoint.

4.3.1 Some useful interval-arithmetic realizations

4.3.1.1. *Integral representation of fixed length.* A first convenient way to represent reals at finite precision is to use integers as an approximate representation.

Definition 4.3.1 (Integral representation of reals)**.** *Let $x \in \mathbf{R}$ be an arbitrary real number and $n \geq 0$ a non-negative integer. Define* an integral representation at accuracy[3] *n as an interval of diameter* 2*:*

$$\underline{x}_n = [X_n - 1, X_n + 1]$$

together with a guarantee that $2^n x$ belongs to $\underline{x}_n$.

This representation is very compact, since it only requires to store the center X_n of the interval using $n + \lceil \log x \rceil$ bits. However, computing with this form of representation is not convenient. As a consequence, we only use it to represent immutable values and we convert to a different representation for computations. The reason for using the interval $[X_n - 1, X_n + 1]$ of diameter 2 rather than $[X_n - 1/2, X_n + 1/2]$ (of diameter 1) is that when $2^n x$ is very close to a half-integer, it remains possible to easily provide a valid value for X_n without computing extraneous bits of the representation of x.

3 We use here the denomination of "accuracy" instead of "precision" to avoid confusions with the floating-point precision as defined in Paragraph 4.3.1.3.

$$\left[\underline{x}^-, \underline{x}^+\right] + \left[\underline{y}^-, \underline{y}^+\right] = \left[\underline{x}^- +^- \underline{y}^-, \underline{x}^+ +^+ \underline{y}^+\right]$$
$$\left[\underline{x}^-, \underline{x}^+\right] - \left[\underline{y}^-, \underline{y}^+\right] = \left[\underline{x}^- -^- \underline{y}^-, \underline{x}^+ -^+ \underline{y}^+\right]$$
$$\left[\underline{x}^-, \underline{x}^+\right] \times \left[\underline{y}^-, \underline{y}^+\right] = \left[\min^-(\rho), \max^+(\rho)\right] \qquad \text{where} \quad \rho = \underline{x}^-\underline{y}^-, \underline{x}^+\underline{y}^-, \underline{x}^-\underline{y}^+, \underline{x}^+\underline{y}^+$$
$$\left[\underline{x}^-, \underline{x}^+\right]^{-1} = \left[\min^-\left(\frac{1}{\underline{x}^+}, \frac{1}{\underline{x}^-}\right), \max^+\left(\frac{1}{\underline{x}^+}, \frac{1}{\underline{x}^-}\right)\right]$$

$\_+^+$, $+^-$ are here respectively the $+$ operator with rounding up or down. The same goes for the $-^+, -^-, \min^-, \max^+$ operators.

Figure 1: Basic arithmetic operators in Interval Arithmetic

4.3.1.2. *Fixed-point representations.* In the context of lattice reduction, it is useful to compute linear combinations with exact integral coefficients. In order to do that with approximate values initially given by centered integral representation, it is possible to use a fixed-point representation.

Definition 4.3.2 (Fixed point representation of reals)**.** *Let $x \in \mathbf{R}$ be an arbitrary real number and $n \geq 0$ a non-negative integer. Define a* fixed-point representation at accuracy n *of radius δ as an interval:*

$$\underline{x}_n = [X_n - \delta, X_n + \delta]$$

together with a guarantee that $2^n x$ belongs to $\underline{x}_n$.

It is easy to add or subtract such intervals by doing the computation on the center and by adding the two radii. It is also easy to multiply by an exact integer by multiplying the center by the integer and the radius by its absolute value. Integral representations are a special case of fixed-point representations, with radius equal to 1.

4.3.1.3. *Floating-point representation.* Another way to handle real values is to use floating point representations of the two bounds of each interval. For example, if we denote by $\lfloor x \rfloor_n$ and $\lceil x \rceil_n$ respectively the largest floating-point number below x and the lowest floating-point number above x written with n bits, the tightest floating-point representation of x with n bits of precision is the interval $I_n(x) = [\lfloor x \rfloor_n, \lceil x \rceil_n]$.

With such a representation, it becomes possible to create a realization of the elementary operations by using careful rounding when computing approximations of the bounds of the resulting interval, as shown in Figure 1. When speaking of the precision of such a representation, we simply refer to the common floating-point precision of the upper and lower bounds.

Once the elementary operations are available, they can be used to implement certified versions of any function that can classically be computed with floating point arithmetic.

4.4 APPROXIMATE LATTICES

The need to reduce lattices given by approximations, especially for number-theoretic applications as been known for long. Using interval arithmetic, it becomes possible to get finer control on the precision required to perform the lattice reduction, even with approximate lattices.

4.4.1 Approximate representation of a positive-definite matrix

A real-valued matrix can easily be represented with the integral representation from Definition 4.3.1, using the same accuracy for all of its entries.

Definition 4.4.1 (Matrix integral representation)**.** *Let $A = (a_{i,j})_{i,j} \in \mathbf{R}^{d\times d}$ be an arbitrary real matrix of dimension d and $n > 0$ be a fixed positive integer. A matrix of intervals*

$$\underline{A}_n = (\underline{a_{i,j}}_n)_{(i,j)\in[1\cdots d]^2},$$

where each $\underline{a_{i,j}}_n$ is an integral representation of $a_{i,j}$ is said to integrally represent *A* at accuracy *n.*

We may omit the subscript n when the accuracy is clear from the context. Given a matrix A, and a matrix $B \in \underline{A}_n$, there exists a unique $d \times d$ matrix Δ with entries in $[-2, 2]$ such that $B = 2^n A + \Delta$.

In particular, we may apply this representation to symmetric matrices. In that case, we obtain the following useful lemma:

Lemma 4.4.1. *Let $S = (s_{i,j})_{i,j} \in \mathcal{S}_d(\mathbf{R})$ be a symmetric matrix of dimension d and $\underline{S}_n$ an integral representation of S at accuracy n. Then, for any symmetric matrix S' in $\underline{S}_n$, we have:*

$$2^n\lambda_d(S) - 2d \leq \lambda_d(S') \leq 2^n\lambda_d(S) + 2d,$$

where $\lambda_d(T)$ denotes the smallest eigenvalue of a d-dimensional symmetric matrix T.

Proof. This is a direct consequence of Weyl's inequalities for Hermitian matrices and of the relation $S' = 2^n S + \Delta$, where Δ is real symmetric with entries in $[-2, 2]$. Note that the eigenvalues of Δ all belong to $[-2d, 2d]$. ∎

4.4.2 Representation of lattices

In order to represent arbitrary lattices, we first need a description of their ambient space. We simply describe the ambient space V of dimension d by providing a basis $\gamma = (\gamma_1, \ldots, \gamma_d)$. Then, the scalar product $\langle\cdot,\cdot\rangle$ on V can be encoded by a Gram matrix $\mathcal{G}_\gamma = \left(\langle\gamma_i, \gamma_j\rangle\right)_{(i,j)\in[1\cdots d]^2}$.

When the Gram matrix $\mathcal{G}_\gamma$ is integral, this already is a standard description of the lattice Γ spanned by γ. This representation appears in particular in [34, Proposition 2.5.3]. We now extend this in order to represent bases and generating families of arbitrary sublattices of Γ. Let Λ be a rank $r \leq d$

sublattice of Γ given by a generating family $\ell = (\ell_1, \ldots, \ell_p)$. Since any vector in ℓ belongs to Γ, it can be expressed with integral coordinates in the basis γ. As a consequence, we can represent ℓ by a $p \times d$ integral matrix L. Moreover, the knowledge of $\mathcal{G}_\gamma$ allows us to easily compute the scalar product of any pair of vectors in Λ.

All this leads to the following definition:

Definition 4.4.2 (Approximate representation of a lattice)**.** *Let $\mathcal{G}_\gamma$ and L be as above and n be a non-negative integer. Denote by G the matrix of centers of an integral representation $\underline{\mathcal{G}_\gamma}_n$ at accuracy n of the Gram matrix $\mathcal{G}_\gamma$. Then the pair $(G, L) \in \mathbf{Z}^{d\times d} \times \mathbf{Z}^{p\times d}$ of integral matrices is said to* represent at accuracy *n the lattice Λ in the basis γ of Γ.*

4.4.2.1. *Computation of the inner product in Interval Arithmetic.* Let a and b be two vectors of Λ described by their vectors A and B of coordinates in the basis γ. We know that:

$$\langle a, b\rangle = A^T \cdot \mathcal{G}_\gamma \cdot B.$$

Thus:

$$2^n\langle a, b\rangle = A^T \cdot G \cdot B + A^T \cdot \Delta \cdot B, \text{ where } |A^T \cdot \Delta \cdot B| \leq \left(\sum_i |A_i|\right)\left(\sum_i |B_i|\right).$$

This directly gives an interval representation of $\langle a, b\rangle$.

4.4.3 Lattice reduction of approximate lattices

Suppose now that the Gram matrix $\mathcal{G}_\gamma = \left(\langle \gamma_i, \gamma_j\rangle\right)_{(i,j)\in[1\cdots d]^2}$ representing the inner product of the ambient space $\Gamma \otimes_{\mathbf{Z}} \mathbf{R}$ in the basis γ is given indirectly by an algorithm or an oracle $\mathcal{O}_\gamma$ that can compute each entry at any desired accuracy. We can restate the definition of a reduced basis in this framework as:

Definition 4.4.3 ((δ, η)-LLL reduction)**.** *Let (δ, η) be admissible LLL parameters. Given an integral matrix $L \in \mathbf{Z}^{p\times d}$ which describes the vectors of a basis of a lattice Λ in the basis γ, we say that $(\mathcal{G}_\gamma, L)$ is a (δ, η)-LLL reduced basis of Λ if and only if there exists an $n_0 > 0$ such that for any $n \geq n_0$ there exists a pair $(\underline{G}_n, L)$, where $\underline{G}_n$ is an integral representation of $\mathcal{G}_\gamma$ at accuracy n, which is a (δ, η)-LLL reduced basis.*

The computational problem associated with reduction theory can then be written as:

Problem (Lattice Reduction for approximate representation)**.** *Let δ, η be admissible LLL parameters. The reduction problem for approximate representation if formally defined as:*

> ***Input:*** *Algorithm—or oracle— computing $\mathcal{G}_\gamma$ at arbitrary precision and an integral matrix $L \in \mathbf{Z}^{p\times d}$ that describes the vectors of a generating family of a lattice Λ in the basis γ.*

Output: *Basis L' of Λ such that $(\mathcal{G}_\gamma, L')$ is a (δ, η)-LLL reduced basis in the sense of Definition 4.4.3.*

Note that using interval arithmetic it suffices to check the (δ, η)-LLL reduction condition at accuracy n_0 to be sure it holds at any larger accuracy. Indeed, an integral representation that satisfies the condition can be refined into a more precise integral representation by scaling up the integer representing the center by an adequate power of two. This refined representation continues to satisfy the condition.

4.4.3.1. *Accuracy of representation and space complexity.* Let $(\underline{G}_n, L)$ be an integral representation of Λ, at accuracy n. Then, the magnitude of the entries of G is 2^n times the magnitude of the entries of $\mathcal{G}_\gamma$. Thus, $\underline{G}_n$ can be encoded using $\mathrm{O}\big(d^2(n + \log\|\mathcal{G}_\gamma\|_{\max})\big)$ bits.

4.5 GENERALIZED LLL REDUCTION WITH INTERVAL ARITHMETIC

In this Section, we adapt lattice reduction algorithms to our setting. More precisely, we represent the information related to Gram-Schmidt vectors by interval arithmetic using a floating-point representation as described in Paragraph 4.3.1.3. For the representation of the lattice itself, we consider two cases: either the underlying Gram matrix is integral, or it is given by an approximate integral representation as in Section 4.4.1. In the latter case, our algorithm also asks for representations with higher accuracy until it is sufficient to yield a reduced basis for the given lattice. The canonical case with the standard Euclidean scalar product is achieved by setting the Gram matrix to the (exact) identity matrix.

4.5.1 Interval Arithmetic L^2 reduction with fixed precision.

We first consider the simplified case where the lattice representation is fixed. It can be either exact or approximate with a given accuracy. In both cases, we fix a basis $\gamma = (\gamma_1, \ldots, \gamma_d)$ and a representation of a lattice Λ in this basis. It is respectively an exact integral representation (G, L) or an approximate representation $(\underline{G}_n, L)$ at accuracy n of $(\mathcal{G}_\gamma, L)$.

4.5.1.1. *Using Interval Arithmetic in LLL.* We now modify the L^2 algorithm of [130] in a few relevant places to make use of interval arithmetic instead of floating-point arithmetic for the Gram-Schmidt-related values. Since the description of the lattice Λ is already using intervals, it seems natural to use interval arithmetic in the lattice reduction algorithm. For completeness, when the input Gram matrix is exact, we make the updates to the Gram-Schmidt orthogonalized matrix used by LLL explicit in the algorithm (except the simple displacements). This also emphasizes a subtle difference with the case of an approximate input Gram matrix. Indeed, in that case, we update the GSO-values but recompute the errors rather than relying on the interval

arithmetic to do it. This is important to gain a fine control on the error growth during updates.

In addition, when using the technique from [135] to be able to deal with lattices given by a generating family instead of a basis, we make a slightly different choice than in [130]. Instead of moving the zero vectors that are encountered during the computation during the reduction to the start of the basis, we simply remove them. Note that with an approximate matrix, if we discover a non-zero vector whose length is given by an interval containing 0, it is not possible to continue the computation. This means that the accuracy of the input is insufficient and we abort. The core modification with interval arithmetic appears while testing the Lovász condition. If it is not possible to decide whether the test is true or false because of interval overlap, we also abort due to lack of precision. To be more precise, when testing the Lovász condition, we also need to check that the corresponding μ coefficient is indeed smaller than η. The reason for this is that, when called with insufficient precision, the Lazy reduction routine may fail to ensure that property.

In addition, if a negative number occurs when computing the norm of a vector, it means that the given Gram matrix is not positive-definite and the algorithm returns an error accordingly.

Algorithm 17 — η-ILazyRed

Input : Initial basis $L = (L_1, \ldots, L_d)$, precomputed (internal) Gram matrix $Gram$, interval matrices $\underline{R}$ and $\underline{M}$, an integer $1 \leq k \leq d$.

Output : Size-reduce the k-th vector of L and update the Gram matrix $Gram$.

done $\leftarrow$ **false**
while *done* $=$ **false do**
  for $j = 1$ **to** $k-1$ **do**
    $\underline{R_{k,j}} \leftarrow$ ConvertToFPInterval$(Gram_{k,j})$
    for $i = 1$ **to** $j-1$ **do** $\underline{R_{k,j}} \leftarrow \underline{R_{k,j}} - \underline{M_{j,i}}\underline{R_{k,i}}$
    $\underline{M_{k,j}} \leftarrow \underline{R_{k,j}} / \underline{R_{j,j}}$;
  end for
  $\underline{s_1^{(k)}} \leftarrow$ ConvertToFPInterval$(Gram_{k,k})$
  for $j = 2$ **to** k **do** $\underline{s_j^{(k)}} \leftarrow \underline{s_{j-1}^{(k)}} - \underline{M_{k,j-1}} \cdot \underline{R_{k,j-1}}$
  $\underline{R_{k,k}} \leftarrow \underline{s_k^{(k)}}$
  $\underline{\tau} \leftarrow (\max_{j<k} \underline{M_{k,j}})$
  ret $\leftarrow (\underline{\tau} \leq \eta)$
  if *ret* $\neq$ **false then** done $\leftarrow$ **true**
  else
    for $i = k-1$ **downto** 1 **do**
      $X_i \leftarrow \eta$-IntervalClosestInteger$(\underline{M_{k,i}})$
      for $j = 1$ **to** $i-1$ **do** $\underline{M_{k,j}} \leftarrow \underline{M_{k,j}} - X_i \underline{M_{i,j}}$
      $L_k \leftarrow L_k - X_i L_i$
      // Update the Gram matrix accordingly
      $Gram_{k,k} \leftarrow Gram_{k,k} - 2X_i Gram_{k,i} + X_i^2 Gram_{i,i}$
      for $j = 1$ **to** i **do** $Gram_{k,j} \leftarrow Gram_{k,j} - X_i Gram_{i,j}$
      for $j = i+1$ **to** $k-1$ **do**
        $Gram_{k,j} \leftarrow Gram_{k,j} - X_i Gram_{j,i}$
      for $j = k+1$ **to** d **do**
        $Gram_{j,k} \leftarrow Gram_{j,k} - X_i Gram_{j,i}$
    end for
  end if
end while

Algorithm 18 – $\tilde{\text{L}}^2$ Algorithm

Parameter : $\delta \in (1/4, 1), \eta \in (1/2, \sqrt{\delta})$ admissible LLL parameters, $\ell \in \mathbf{N}$ the internal precision used for floating-point representation.
Input : Exact representation (G, L) or approximate representation $(\underline{G_n}, L)$ of a lattice given by p generating vectors in dimension d.
Output : A (δ, η) LLL-reduced basis L' (with $\dim(L)$ vectors).

$k \leftarrow 2$
// Compute the Gram matrix of the basis represented by L
for $i = 1$ **to** p **for** $j = 1$ **to** i **do**
  if *Exact* **then** $GramL_{i,j} \leftarrow L_i^T G L_j$
  else $GramL_{i,j} \leftarrow$ Interval of center $L_i^T G_n L_j$ and radius $\|L_i\|_1 \|L_j\|_1$
end for
$\underline{R_{1,1}} \leftarrow$ ConvertToFPInterval$(GramL_{1,1})$
while $k \leq p$ **do**
  // Size-reduce L_k with interval on the family $(L_1, \ldots, L_{k-1})$
  η-ILazyRed(k, *Exact*)
  if *Exact* = **false then for** $j = 1$ **to** k **do**
    Update radius of $GramL_{k,j}$ to $\|L_k\|_1 \|L_j\|_1$ (rounded up with ℓ significant bits)
  end for
  $k' \leftarrow k$
  while $k \geq 2$ **do**
    ret $\leftarrow \left(\underline{M_{k',k-1}} \leq \eta\right)$ and $\left(\underline{\delta} \cdot \underline{R_{k-1,k-1}} > \underline{s_{k-1}^{(k')}}\right)$
    if *ret* = **true then** $k \leftarrow k - 1$
    else if *ret* = **false then break**
    else return ErrorPrecision
  end while
  if $k \neq k'$ **then**
    for $i = 1$ **to** $k - 1$ **do** $\underline{M_{k,i}} \leftarrow \underline{M_{k',i}}$; $\underline{R_{k,i}} \leftarrow \underline{R_{k',i}}$
    $\underline{R_{k,k}} \leftarrow \underline{s_k^{k'}}$
    $L_{tmp} \leftarrow L_{k'}$
    for $i = k'$ **downto** $k + 1$ **do** $L_i \leftarrow L_{i-1}$
    $L_k \leftarrow L_{tmp}$; Move values in $GramL$ accordingly
  else
    $\underline{R_{k,k}} \leftarrow \underline{s_k^{(k')}}$
    if $0 \in \underline{R_{k,k}}$ *and* $L_k \neq 0$ **then return** ErrorAccuracy
    if $\underline{R_{k,k}} < 0$ **then return** ErrorNonPosDefinite
  end if
  if $L_k = 0$ **then**
    // Remove zero vector from L
    for $i = k$ **to** $p - 1$ **do** $L_i \leftarrow L_{i+1}$
    $p \leftarrow p - 1$; $k \leftarrow k - 1$; Move values in $GramL$ accordingly
  end if
  $k \leftarrow \max(k + 1, 2)$
end while
return (L)

4.5.1.2. *Internal precision in the exact-input case.* For the classical L^2 algorithm, Paragraph 4.2.2.1 states that the precision that is needed for the computations only depends on the dimension of the lattice. It is natural to ask a similar question about the algorithm $\tilde{\text{L}^2}$: can the required internal accuracy be bounded independently of the entries appearing in the matrices G and L. When G is exact, i.e., integral, the adaptation is straightforward and we obtain the following result.

Theorem 4.5.1. *Let (δ, η) be admissible LLL parameters. Let $c > \log \frac{(1+\eta)^2}{\delta-\eta^2}$ and let $(\Lambda, \langle\cdot,\cdot\rangle)$ denote a rank-d lattice, exactly described by the pair (G, L). Let B denotes the maximum entry in absolute value in $L^T G L$. Then, the $\tilde{\text{L}^2}$ of Algorithm 18 used with $\ell = cd + o(d)$ outputs a (δ, η)-LLL-reduced basis in time $O\big(d^3 \log B(d + \log B)\mathcal{M}(d)\big)$. Furthermore, if τ denotes the number of main loop iterations, the running time is $O(d(\tau + d \log dB)(d + \log B)\mathcal{M}(d))$.*

In fact, the bound on ℓ is made explicit in [130]. More precisely, it states that for any arbitrary $C > 0$ and an $\epsilon \in]0, 1/2]$, it suffices to have:

$$\ell \geq 10 + 2\log_2 d - \log_2 \min(\epsilon, \eta - 1/2) + d(C + \log_2 \rho),$$

where $\rho = \frac{(1+\eta)^2+\epsilon}{\delta-\eta^2}$. For example, choosing $C = \epsilon = \eta - 1/2$ it suffices to have:

$$\ell \geq T(d, \delta, \eta) = 10 + 2\log_2 d - \log_2(\eta - 1/2) + (\eta - 1/2 + \log_2 \rho)\, d.$$

When δ is close to 1 and η to $1/2$, the constant before d becomes smaller than 1.6.

4.5.1.3. *Dealing with approximate inputs.* When dealing with lattices given in an approximate form, i.e., by a representation $(\underline{G_n}, L)$ at accuracy n of $(\mathcal{G}_\gamma, L)$, the analysis of the algorithms differs in three main places:

- When bounding the number of rounds τ, we can no longer assume that the potential is an integer. As a consequence, in order to keep a polynomial bound on τ, we need to provide a lower bound on the possible values of the potential, rather than rely on the trivial lower bound of 1 for an integral-valued potential.
- Since the notion of LLL-reduction is only well-defined for a positive definite G, we need to make sure that $\underline{G_n}$ is positive-definite during the algorithm. Otherwise, it should output an error; Algorithm 18 returns an error that $\underline{G_n}$ is incorrect whenever it encounters a vector with a negative norm.
- When $\underline{G_n}$ is approximate, the scalar products between lattice vectors can no longer be exactly computed. Thus, we need to able to make sure that the errors are small enough to be compatible with the inner precision used for Gram-Schmidt values. At first glance, this might seem easy. However, when using update formulas to avoid recomputation of scalar products, the estimates on errors provided by interval

arithmetic can grow quite quickly. In fact,it would prevent the update strategy from working. The key insight is to remark that since the centers of the intervals are represented by integers, any computation on them is exact and we can use update formulas to compute them. However, it is essential to recompute the radii of the intervals, i.e., the errors, to prevent them from growing too quickly.

Number of rounds. Since interval arithmetic allows up to emulate exact computations as long as no failures are detected, we can analyze the number of rounds by assuming that all computations on non-integral values are done using an exact arithmetic oracle. In this context, the number of rounds can be studied by considering the potential as usual. Remember that the initial setting where LLL operates on a basis the potential is defined as

$$\Pi(B) = \prod_{i=1}^{d} \operatorname{covol}\left(B_{[1\ldots i]}\right).$$

The key argument is that it decreases by a multiplicative factor whenever an exchange is performed.

However, in our context, the starting upper bound and the ending lower bound are different from the integer lattice setting. The initial upper bound needs to account from the presence of the positive definite matrix. So if the lattice is described by a pair $(\mathcal{G}_\gamma, L)$ the upper bound becomes:

$$\Pi(B)^2 \leq \left(d^2 \|\mathcal{G}_\gamma\|_{\max} \|L\|_{\max}^2\right)^{d(d+1)/2}.$$

More importantly, it is no longer possible to claim that the potential is an integer. Instead, we derive a lower bound by considering the smallest eigenvalue of $\mathcal{G}_\gamma$ and find:

$$\Pi(B)^2 \geq \lambda_d(\mathcal{G}_\gamma)^{d(d+1)/2}.$$

As a consequence, if we let τ denote the number of rounds of the algorithm, we can conclude that:

$$\tau \leq \mathrm{O}\big(d^2(\log(\|L\|_{\max}) + \log(\|\mathcal{G}_\gamma\|_{\max}/\lambda_d(\mathcal{G}_\gamma)) + \log(d)\big).$$

When the lattice is given by a generating family L rather than a basis B, we need a slightly different invariant. Following [130], we define d_i to be the product of the first i non-zero values $\|b_j^*\|$. Note that they are not necessarily consecutive, since zeroes may occur anywhere. We then let:

$$\Pi'(L) = \left(\prod_{i=1}^{\dim L} d_i\right) \cdot \left(\prod_{i, b_i^* = 0} 2^i\right).$$

This generalized potential is needed for the proof of Theorem 4.5.2. Note that, for lattices given by a basis, the two definitions coincide.

Necessary accuracy for the scalar products. In order to preserve the correctness of the algorithm when computing with internal precision ℓ, we need to check that all conversions of scalar product values, using the calls to ConvertToFPInterval in Algorithm 17 and Algorithm 18, have sufficient precision. For a pair of lattice elements, described by vectors L_i and L_j, the relative precision on the value of their scalar product is:

$$\frac{\|L_i\|_1\|L_j\|_1}{|L_i^T G_n L_j|}.$$

When the vectors are close to orthogonal with respect to the scalar product given by G_n, the error can be arbitrarily large. However, by carefully following the analysis of Theorem 3 in [130, Section 4.1], we can show that this Theorem remains true in our context. This suffices to ensure the correctness part of Theorem 5 of [130]. The first check is to verify that quantity called err_1 in the proof of the Theorem remains upper bounded by $2^{-\ell}$. Since the value is defined as the error on the scalar product of the vectors number i and 1 divided by the norm of the first vector, we have:

$$err_1 \leq \frac{\|L_i\|_1\|L_1\|_1}{|L_1^T G_n L_1|} \leq \frac{\max_i \|L_i\|_1^2}{\lambda_d(G_n)} \leq \frac{d \max_i \|L_i\|^2}{\lambda_d(G_n)} \leq \frac{d \max_i \|b_i\|^2}{\lambda_d(G_n)^2}.$$

Thus:

$$err_1 \leq \frac{d^3\|G_n\|_{\max}\|L\|_{\max}^2}{\lambda_d(G_n)^2} \leq \frac{d^3(2^n\|\mathcal{G}_\gamma\|_{\max}+1)\|L\|_{\max}^2}{(2^n\lambda_d(\mathcal{G}_\gamma)-2d)^2}.$$

As a consequence, it suffices to have:

$$n \geq \ell + \mathrm{O}(\log(\|L\|_{\max}) + \log(\|\mathcal{G}_\gamma\|_{\max}/\lambda_d(\mathcal{G}_\gamma)) + \log(d)).$$

L^2 with approximate inputs. To complete the above properties on the number of rounds and necessary accuracy, it suffices to remark that the only additional line of code in the approximate L^2 is the recomputation of interval radii on line 10. Since it suffices to know the ℓ high-order bits of the values, this recomputation can fully be done using arithmetic on ℓ. Indeed, during the computations of $\|L_i\|_1$ no cancellation occurs. As a consequence, we get the following adaptation of Theorem 4.5.1. For completeness, we give here the case where the lattice is initially given by a generating family of p vectors, has rank d and lives in an ambient space of dimension D.

Theorem 4.5.2. *Let (δ, η) be such that $1/4 < \delta < 1$ and $1/2 < \eta < \sqrt{\delta}$. Let $c > \log \frac{(1+\eta)^2}{\delta-\eta^2}$. Assume that we are given as input $(\Lambda, \langle\cdot,\cdot\rangle)$ a rank-d lattice $(\mathcal{G}, L)$ described by $p \geq d$ generating vectors in a ambient space of dimension $D \geq d$. Further assume that it is approximately represented at accurary N by the pair $(\underline{G_N}, L)$ and let B denote the maximum entry in absolute value in $L^T\mathcal{G}L$. Let $\ell = cd + o(d)$ and*

$$N \geq \ell + \log(B/\lambda_D(\mathcal{G})) + \log(d).$$

Then, the $\tilde{\text{L}}^2$ *of Algorithm 18 outputs a* (δ, η)*-LLL-reduced basis in time*

$$O\big(DN\big(d^2N + p(p-d)\big)\mathcal{M}(d)\big).$$

Furthermore, if τ *denotes the number of main loop iterations, the running time is* $O(DN(dN+\tau)\mathcal{M}(d))$.

4.5.2 L^2 reduction with adaptive precision and accuracy.

4.5.2.1. *Adaptive precision.* Since by construction the $\tilde{\text{L}}^2$ Algorithm can detect that the choice for internal precision ℓ is insufficient to correctly reduce the lattice Λ. The procedure can be wrapped in a loop that geometrically increases precision ℓ after each unsuccessful iteration. This yields an *adaptive precision* reduction algorithm ADAPTIVE-LLL. Since the complexity of floating-point multiplication is superlinear, the use of a geometric precision growth guarantees that the total complexity of this lattice reduction is asymptotically dominated by its final iteration.[4]

Moreover, the cost of operations in the floating-point realization of interval arithmetic is at most four times the cost of floating-point arithmetic at the same precision. Depending on the internal representation used, this constant can even be improved. As a consequence, for lattices that can be reduced with a low-enough precision, it can be faster to use interval arithmetic than floating-point arithmetic with the precision required by the bound from Paragraph 4.2.2.1.

4.5.2.2. *Adaptive accuracy.* We now turn to the setting of Section 4.4.3, where an algorithm or oracle $\mathcal{O}_\gamma$ can output an integral representation of the Gram matrix $\mathcal{G}_\gamma = \big(\langle \gamma_i, \gamma_j \rangle\big)_{(i,j)\in[1\cdots r]^2}$ at arbitrary accuracy n. In that context, we need to determine both the necessary accuracy and internal precision. When running Algorithm 18 with some given accuracy and precision, three outcomes are possible:

- Either the reduction terminates in which case the lattice is LLL-reduced, which implies that both accuracy and precision are sufficient.
- The Lovász condition fails to be tested correctly, which indicates an insufficient precision. In that case, we need to test whether the precision is lower than theoretical bound $T(d, \delta, \eta)$ given after Theorem 4.5.1 or not. In the latter case, we know that the accuracy needs to be increased.
- The algorithm detects a non-zero vector whose norm is given by an interval containing 0. This directly indicates insufficient accuracy.

Depending on the result of Algorithm 18, we increase the precision or the accuracy and restart. The corresponding pseudo-code is given in Algorithm 19. Since the precision and accuracy both follow a geometric growth,

4 In practice, for lattices of rank few hundreds it appears nonetheless that the computational cost of the previous iterations lies between 20% and 40% of the total cost.

the computation is dominated by its final iteration. In particular, we may use the complexity bound given by Theorem 4.5.2.

Note that when we increase the accuracy in Algorithm 19, we also reset the precision to its minimal value. This is a matter of preference that doesn't affect the asymptotic complexity. In practice, it seems to be preferable.

It is important to note that we do need to precompute the eigenvalues of the Gram matrix, since Algorithm 19 automatically detects the needed accuracy.

Algorithm 19 — ADAPTIVE-LLL

Parameter : $\delta \in (1/4, 1), \eta \in (1/2, \sqrt{\delta})$, $\ell_0 \in \mathbf{N}$ initial precision of the algorithm for floating-point representation, n_0 initial accuracy for representing the scalar product, $g > 1$ geometric growth factor.
Input : γ a basis of a lattice $(\Gamma, \langle \cdot, \cdot \rangle)$, and $\mathcal{O}_\gamma(n)$ an oracle that compute the integral representation of the inner product $\langle \cdot, \cdot \rangle$ at accuracy n.
Input : A generating family represented by L in γ of a sublattice $\Lambda \subset \Gamma$.
Output : A (δ, η) LLL-reduced basis of Λ represented as $L' \in \mathbf{Z}^{\mathrm{rk}(\Lambda) \times \mathrm{rk}(\Lambda)}$.

// Set initial values for accuracy and precision
// $\mathrm{T}(d, \delta, \eta)$ is the theoretical bound given after Theorem 4.5.1
$\ell \leftarrow \ell_0$
$n \leftarrow n_0$
$G \leftarrow \mathcal{O}_\gamma(n)$
succeed $\leftarrow$ **false**
repeat
  retcode $\leftarrow \tilde{\mathrm{L}^2}(G, L)$
  if *retcode=ErrorNonPosDefinite* **then return** ErrorNonPosDefinite
  if *retcode=OK* **then** succeed $\leftarrow$ **true**
  else if *retcode=ErrorPrecision* **then**
    $\ell' \leftarrow \ell$
    $\ell \leftarrow \min(\lceil g\,\ell \rceil, T(d, \delta, \eta), n)$
    if $\ell' = \ell$ **then** retcode $\leftarrow$ ErrorAccuracy
  end if
  if *retcode=ErrorAccuracy* **then**
    $\ell \leftarrow \ell_0$
    $n \leftarrow \lceil g\,n \rceil$
    $G \leftarrow \mathcal{O}_\gamma(n)$
  end if
until *succeed* = **true**
return L

4.5.3 Possible generalizations and practical impact

The adaptative strategy we describe for LLL can be generalized to other lattice reduction algorithms. In particular, enumeration algorithms are possible within our framework, which allows the implementation of the BKZ algorithm of [145]. It would be interesting to study a generalization to sieving techniques to adapt them to approximate lattices. This framework can straightforwardly be applied to the generalization of Napias' algorithm of [30], as it relies on floating-point computations to handle internal values. It would provide a speed-up based on the use of a smaller internal precision than the worst-case bound used. For instance, used on `fpLLL` it allows us to halve the running-time of reductions.

4.6 BACK TO THE REDUCTION OF ALGEBRAIC LATTICES

We now go back to our matter of interest and detail the application of this interval arithmetic framework to the reduction of algebraic lattices in algorithmic number theory. As a warm-up, we start by the case of ideal lattices, as introduced in Paragraph 3.3.1.3. In all of the following, suppose that $\mathbf{L}$ is a number field of degree n over $\mathbf{Q}$ and that we are given an integral basis $(w_1, \ldots, w_n)$ of $\mathscr{O}_\mathbf{L}$. As a warmup, we start by exposing the reduction process for ideal lattices.

4.6.1 Lattice reduction for ideals.

With the above notations, we can directly use our lattice reduction algorithm to reduce an ideal lattice. More precisely, given a two-element representation of $\mathfrak{a}$ by $\alpha \in \mathbf{L}$ and $\beta \in \mathbf{L}$, we proceed as follows:

1. Define the Gram matrix $\mathcal{G}_w$ with entries $\langle w_i, w_j \rangle_\sigma$. It can be computed to any desired precision from approximations of the roots of P. The roots themselves can be computed, using, for example, the Gourdon-Schönhage algorithm [69].
2. Let L be the matrix formed of the coordinates of $(\alpha w_1, \ldots, \alpha w_n)$ and $(\beta w_1, \ldots, \beta w_n)$ in the basis $(w_1, \ldots, w_n)$.
3. Directly apply Algorithm 19 to $(\mathcal{G}_w, L)$.

The same thing can be done, *mutatis mudantis*, for an ideal described by a $\mathbf{Z}$-basis.

4.6.2 Lattice reduction for algebraic lattices.

Let us take (Λ, A) an algebraic lattice of rank d over $\mathscr{O}_\mathbf{L}$. Suppose that $\Lambda = \alpha_1\mathfrak{a}_1 \oplus \alpha_2\mathfrak{a}_2 \oplus \cdots \oplus \alpha_d\mathfrak{a}_d$ is a decomposition of the lattice in pseudo-basis. We perform the computation in the same way as for ideal lattices:

1. Define the Gram matrix $\mathcal{G}_w$ with entries $\langle w_i, Aw_j \rangle_\sigma$. It can be computed to any desired precision from approximations of the roots of P. In the case where the Hermitian metric is given as a Humbert form $(A_\sigma)_{\sigma:\mathbf{L}\to\mathbf{C}}$, we compute the matrix $\mathcal{G}_w$ as: $\sum_{\sigma:\mathbf{L}\to\mathbf{C}} \sigma(w_i)^T A_\sigma \sigma(w_j)$.

2. For each fractional ideal $\mathfrak{a}_i$, denote by $(x_1^{(i)}, \ldots, x_n^{(i)})$. Let now L be the matrix formed of the (rational) coordinates of

$$(\alpha_1 x_1^{(1)}, \ldots, \alpha_1 x_n^{(1)}), (\alpha_2 x_2^{(2)}, \ldots, \alpha_2 x_n^{(2)}), \ldots, (\alpha_d x_1^{(d)}, \ldots, \alpha_d x_n^{(d)})$$

   in the basis $(w_1, \ldots, w_n)$.

3. Apply Algorithm 19 to $(\mathcal{G}_w, L)$.

A well-known special case. For some number fields, the Gram matrix $\mathcal{G}_w$ is integral. In that case, the use of Algorithm 19 isn't necessary and one can directly work with an exact lattice. This is described for the special case of reducing the full lattice corresponding to the ring of integers in [12, Section 4.2] for totally real fields. It can be generalized to CM-fields, since they satisfy the same essential property of having an integral Gram matrix. The same application is also discussed in [34, Section 4.4.2].

Non integral case. For the general case where the Gram matrix is real, [12] proposes to multiply by 2^e and round to the closest integer. It also gives a bound on the necessary accuracy e as the logarithm of (the inverse of) the smallest diagonal entry in the Cholesky decomposition of the Gram matrix. In some sense, this is similar to our approach. But the bound can only be oriented at runtime. However, without any auxiliary information on this coefficient, it is proposed to continue *increasing e as long as it is deemed unsatisfactory.*

By contrast, termination of our algorithm guarantees that lattice reduction is completed and that the output basis is LLL-reduced.

5

TOWARDS A FAST REDUCTION OF ALGEBRAIC LATTICES

In the previous chapter, we devised a method to soundly reduce algebraic lattices using Interval Arithmetic. This corresponds to forgetting the algebraic structure of the module and running a reduction algorithm on it. But the image over $\mathbf{Z}$ of a rank d algebraic lattice is of rank $d \times n$, where n is the degree of field inside which we are working initially. Hence, even in the case where the lattice is of small rank, the reduction can be very costly as the actual dimension over $\mathbf{Z}$ might be large. This process is forgetful of the algebraic specificities of the base ring. But these properties translate into symmetries over modules, as they are very structured. Consequently, the above-mentioned reduction cannot take these symmetries into account. Thus, it is natural to wonder if it is possible to *exploit* the algebraic structure of the fields to speed up the reduction. In this chapter, we introduce a framework of techniques to provide fast polynomial-time algorithms for reducing algebraic lattices. We instantiate this framework for cyclotomic fields and discuss its generalization to arbitrary number fields. This is the first time, up to our knowledge, that we can take into account the special structure of ideal and module lattices over cyclotomic fields answering the question of Peikert reported in Section 3.4.

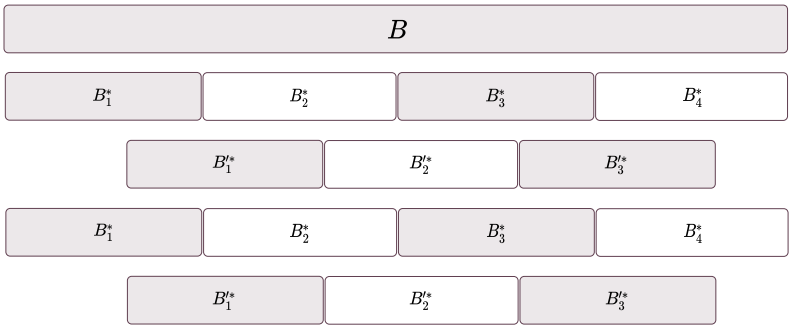
Before diving into the detail of the reduction algorithms, we give a brief overview of the key ideas which are going to be used and combined.

TECHNICAL FRAMEWORK

The core design principles of our framework to provide fast polynomial-time algorithms for reducing algebraic lattices defined over cyclotomic fields are:

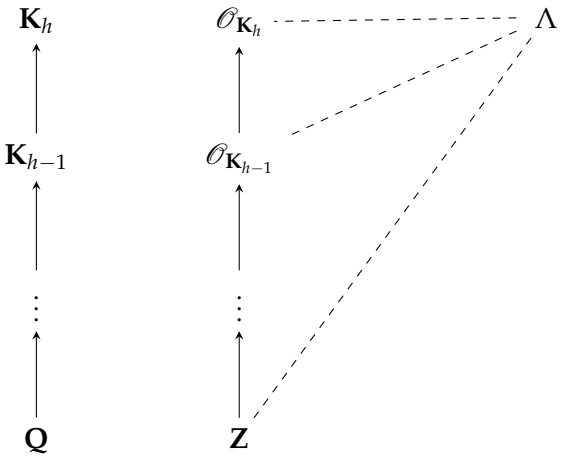
> **A recursive strategy on the rank:** The reduction of a rank d lattice is performed recursively on large blocks. Instead of relying on a local (LLL-like) strategy consisting in choosing the first (or an arbitrary) block for which some progress can be made, we systematically perform the reduction of the blocks. This global process is somewhat similar to the ironing out strategies of BKZ-like reductions or to the fast variant of LLL of Neumaier and Stehlé [128], where successive passes of local reductions are made on the *whole* basis to gradually improve its reduceness. However, we differ from the iterative design à la BKZ as we shift the blocks between odd and even steps to mix all basis vectors as in the early parallelized versions of LLL of Villard [161]. A generic instance of two successive passes of our strategy is given in

the following:



The basis B is here sundered in four chunks B_1, B_2, B_3, B_4 of length $|B|/4$. The reduction process will start by reducing (possibly at the same time) the first chunk $B_1^* = B_1$, the projection B_2^* of the second one orthogonally to B_1, the projection B_3^* of the third one orthogonally of $B_1 \| B_2$ and so on. When this pass is over, the same process starts again, but this time on shifted blocks (i.e. the first block B_1' starts with the vector $|B|/8$ and is of length $|B|/4$). Hence, the rank of the lattices which are called recursively decreases until we reach rank 2 lattices, where we can use a fast reduction like Schönhage's algorithm [148].

A recursive strategy on the degree of the field: Suppose that we are given a tower of number fields $\mathbf{K}_0 \subset \mathbf{K}_1 \subset \ldots \subset \mathbf{K}_h$. Let Λ be an algebraic lattice defined over the ring of integers of the upper field $\mathbf{K}_h$. We can look at Λ as an algebraic lattice defined over the field right under, that is $\mathbf{K}_{h-1}$.



Such an identification is possible at the cost of increasing the rank of the lattice: the rank of Λ seen over $\mathbf{K}_{h-1}$ is exactly $[\mathbf{K}_h : \mathbf{K}_{h-1}]$ times its rank over $\mathbf{K}_h$. Then we make use of the recursive design over the rank, introduced above, to reduce this problem into numerous

instances of reduction of rank two lattices over $\mathbf{K}_{h-1}$. Each of them can be seen over $\mathbf{K}_{h-2}$, inviting us to pursue this descent until we get to the bottom of the tower and are now reducing lattices over $\mathbf{Z}$, that is, Euclidean lattices.

A generic use of symplectic structures in number fields: A Euclidean space is a vector space endowed with a positive definite symmetric bilinear form acting on it. Replacing this form by an antisymmetric one yields the notion of *symplectic space*. Lattices embedded in symplectic spaces have additional symmetries that can be exploited to (roughly) halve the cost of the reduction. We prove that we can define a recursive symplectic structure over a tower of number fields. As a consequence we can halve the running time of the reduction at *each* level of the recursion tree, yielding significant asymptotic speedups on the overall reduction.

A (controlled) low precision reduction: We use approximations instead of exact computations, which corresponds to reducing the projected sublattices with only the most significant bits of their basis. A careful analysis of the precision required to ensure a global reduction gains a factor up to d depending on the condition number of the initial basis, where d is the rank of the lattice we want to reduce. Furthermore, we can show that the precision needed will significantly decrease during *some* recursive calls, up to a factor of d once again.

A fast and generic algorithmic for the log-unit lattice: During the reduction of algebraic lattice, we need to balance the size of the Archimedean embeddings of elements to avoid a blow-up of the precision used. This can be done by carefully multiplying the considered quantities by units of the field, yielding a decoding problem in the so-called *log-unit lattice* of cyclotomic fields. We prove that these results can be achieved within quasilinear running time for any cyclotomic field.

5.1 FAST UNIT-ROUNDING IN CYCLOTOMICS FIELDS

We start by proving the following theorem, as it will appear recurrently in all the following parts of this chapter:

Theorem 5.1.1. *Let $\mathbf{L}$ be the cyclotomic field of conductor f. There is a quasi-linear randomized algorithm that given any element in $x \in (\mathbf{R} \otimes \mathbf{L})^\times$ finds a unit $u \in \mathscr{O}_{\mathbf{L}}^\times$ such that for any field embedding $\sigma : \mathbf{L} \to \mathbf{C}$ we have*

$$\sigma\left(xu^{-1}\right) = 2^{O\left(\sqrt{f \log f}\right)} N_{\mathbf{L}/\mathbf{Q}}(x)^{\frac{1}{\varphi(f)}}.$$

Remark. *Recall that $\frac{f}{\varphi(f)} = O(\log\log f)$, then denoting by $n = \varphi(n)$ the dimension of $\mathbf{L}$, we then shall use the bound $2^{O\left(\sqrt{n \log n \log\log n}\right)} N_{\mathbf{L}/\mathbf{Q}}(x)^{\frac{1}{n}}$, in the result of Theorem 5.1.1.*

We call **Unit** the corresponding program.

Given an arbitrary element u in $\mathbf{L}$, **Unit** allows to systematically and efficiently find a unit in the field which *balance* the Archimedean embeddings of u. This process is the cornerstone of all the algorithms in this manuscript using a descent of an algebraic lattice, as it enable a fine-grained control of the size of the direct image of the elements. This allows in particular to use small precision approximation of the actual values computed.

Example. *Let $\mathbf{L} = \mathbf{Q}[\zeta]$ be the cyclotomic field of conductor 16, and thus of degree 8. Let us consider the embeddings $(\sigma_\alpha)_{\alpha\in(\mathbf{Z}/16\mathbf{Z})^\times}$ of $\mathbf{L}$, defined by sending ζ to ζ^α. These embeddings are grouped by conjugates (namely σ_1 and σ_{15}, σ_3 and σ_{13}, σ_5 and σ_{11}, σ_7 and σ_9. Take for instance the element $x \in \mathbf{L}$ given by:*

$$x = \frac{65210}{3}\zeta^7 + \frac{78658}{3}\zeta^6 - 41412\zeta^5 + \frac{16567}{3}\zeta^4 + 36970\zeta^3 - \frac{100235}{3}\zeta^2 - \frac{145843}{12}\zeta + \frac{86961}{2}.$$

As we have by direct computation $N_{\mathbf{L}/\mathbf{Q}}(x) = \frac{1696380897806689}{429981696} \approx 3945239.79$, we could wonder on how balanced are the norm of the embeddings around their geometric mean $N_{\mathbf{L}/\mathbf{Q}}(x)^{\frac{1}{8}} \approx 6.6758$. A direct computation of the modules of the embeddings reveals that:

$$\begin{aligned}
|\sigma_1(x)| &= |\sigma_{15}(x)| \approx 2771.189\\
|\sigma_3(x)| &= |\sigma_{13}(x)| \approx 1.558406 \times 10^{-08}\\
|\sigma_5(x)| &= |\sigma_{11}(x)| \approx 172334.9\\
|\sigma_7(x)| &= |\sigma_9(x)| \approx 266.8642.
\end{aligned}$$

*indicating that these values are far from being concentrated around their mean. However when multiplied by the unit u^{-1} for $u = -5080\zeta^7 + 6664\zeta^6 - 6664\zeta^4 + 5080\zeta^3 + 2856\zeta^2 - 7361\zeta + 2856 \in \mathscr{O}_{\mathbf{L}}^\times$, obtained by the algorithm **Unit**, we obtain a far better balance:*

$$\begin{aligned}
|\sigma_1(x)| &= |\sigma_{15}(x)| \approx 7.83729\\
|\sigma_3(x)| &= |\sigma_{13}(x)| \approx 7.33868\\
|\sigma_5(x)| &= |\sigma_{11}(x)| \approx 5.93346\\
|\sigma_7(x)| &= |\sigma_9(x)| \approx 5.82028.
\end{aligned}$$

In particular, the algebraic norm of x and xu^{-1} are equal, but the canonical norm of x ($\approx$ 172357.38) is orders of magnitude larger than the canonical norm of xu^{-1} ($\approx$ 13.57795).

The proof of this theorem relies on a randomized rounding in the so-called log-unit lattice of the field. We perform here a novel analysis of the algorithm of [42] allowing a faster running time and we extend their result for *arbitrary* cyclotomic fields.

5.1.1 Prime power-case

As a starter, we prove that the techniques of [42] allows to do the unit-rounding in prime-power cyclotomic fields in quasi-linear time. Formally we aim at proving the following:

Theorem 5.1.2. *Let* $\mathbf{L}$ *be the cyclotomic field of prime power conductor* f*. There is a quasi-linear randomized algorithm that given any element in* $x \in (\mathbf{R} \otimes \mathbf{L})^\times$ *finds a unit* $u \in \mathscr{O}_{\mathbf{L}}^\times$ *such that for any field embedding* $\sigma : \mathbf{L} \to \mathbf{C}$ *we have*

$$\sigma\left(xu^{-1}\right) = 2^{O\left(\sqrt{f \log f}\right)} N_{\mathbf{L}/\mathbf{Q}}(x)^{\frac{1}{\varphi(f)}}.$$

Compared to [42], there are two differences with the treatment proposed here: on the one hand we use fast arithmetic of the involved objects—namely Fourier-based multiplication in an abelian group-ring—and on the other hand we increase the success probability by using a better bound by the classical Berry-Esseen theorem, as it was hinted in their seventh footnote.

5.1.1.1. *Recall on the probability notions used in the proof.* Before diving in the proof of Theorem 5.1.2, let us recall the basis notions of probability theory we are using, namely subgaussians variables and the Berry-Esseen theorem.

On subgaussian random variables. The notion of subgaussian distribution goes back to the work of Kahane in [92], and encompasses a large family of real distributions with very convenient properties similar to the normal law.

Definition 5.1.1. *A real random variable* X *is said to be* τ-subgaussian *for some* $\tau > 0$ *if the following bound holds for all* $s \in \mathbf{R}$*:*

$$\mathbf{E}\left[\exp(sX)\right] \leq \exp\left(\frac{\tau^2 s^2}{2}\right). \tag{5.1}$$

A τ*-subgaussian probability distribution is in an analogous manner.*

Lemma 5.1.1. *A* τ*-subgaussian random variable* X *satisfies*

$$\mathbf{E}[X] = 0.$$

Proof. Follows from the Taylor expansion at 0 of $\mathbf{E}[\exp(sX)] = 1 + s\mathbf{E}[X] + O(s^2)$. ∎

The main property of subgaussian distributions is that they satisfy a Gaussian-like tail bound.

Lemma 5.1.2. *Let* X *be a* τ*-subgaussian distribution. For all* $t > 0$*, we have*

$$\Pr[X > t] \leq \exp\left(-\frac{t^2}{2\tau^2}\right). \tag{5.2}$$

Proof. Fix $t > 0$. For all $s \in \mathbf{R}$ we have, by Markov's inequality:

$$\Pr[X > t] = \Pr[\exp(sX) > \exp(st)] \leq \frac{\mathbf{E}[\exp(sX)]}{\exp(st)}$$

since the exponential is positive. Using that X is τ-subgaussian, Equation 5.1 gives:

$$\Pr[X > t] \leq \exp\left(\frac{s^2\tau^2}{2} - st\right)$$

and the right-hand side is minimal for $s = t/\tau^2$, entailing the announced result. ∎

Many usual distributions over $\mathbf{Z}$ or $\mathbf{R}$ are subgaussian. This is in particular the case for distributions with finite supports and zero mean.

The Berry-Esseen approximation theorem. The Berry-Esseen theorem provides a quantitative estimates of the rate of convergence towards the normal distribution, as showing that the cumulative function (CDF) of the probability distribution of the scaled mean of a random sample converges to Φ at a rate inversely proportional to the square root of the number of samples. More formally we have:

Theorem 5.1.3. *There exists a positive $C < 0.5$ such that if $X_1, X_2, \cdots, X_n$ are independent and identically distributed random variables with zero mean, satisfying $\mathbf{E}(X_1^2) = \sigma^2 > 0$, $\mathbf{E}(|X_1|^3) = \rho$, and by setting*

$$Y_n = \frac{X_1 + X_2 + \cdots + X_n}{n}$$

the sample mean, with F_n the cumulative distribution function of $\frac{Y_n\sqrt{n}}{\sigma}$ and Φ the cumulative distribution function of the standard normal distribution, then for all x and n we have,

$$|F_n(x) - \Phi(x)| \leq \frac{C\rho}{\sigma^3\sqrt{n}}$$

5.1.1.2. *Going back on the rounding problem.* We now fix a cyclotomic field $\mathbf{L} = \mathbf{Q}[\zeta_f]$ with prime power-conductor f and degree n. We recall that in $\mathbf{L}$, the cyclotomic units are easily described:

Lemma 5.1.3 (Lemma 8.1 of [163])**.** *Let f be a prime power, then the group of cyclotomic units is generated by $\pm\zeta_f$ and $\frac{\zeta_f^a - 1}{\zeta_f - 1}$ for $a \in (\mathbf{Z}/f\mathbf{Z})^\times$.*

We first provide a convenient description of the cyclotomic units as an orbit of the element $\zeta_f - 1$ under the action of its Galois group.

5.1.1.3. *Log-embedding and action of* $(\mathbf{Z}/f\mathbf{Z})^\times/\{-1,+1\}$. Define the Log embedding to be the coefficient-wise composition of the real logarithm with the absolute value of Archimedean embeddings:

$$\mathrm{Log}: \left| \begin{array}{ccc} \mathbf{L} & \longrightarrow & \mathbf{R}^{\frac{n}{2}} \\ \alpha & \longmapsto & [\log(|\sigma_i(\alpha)|)]_{i\in G} \end{array} \right. ,$$

where the embeddings are paired by conjugates and listed by the group $G = (\mathbf{Z}/f\mathbf{Z})^\times/\{-1,+1\}$. The image of the unit multiplicative group $\mathscr{O}_{\mathbf{L}}^\times$ is a full rank lattice by Dirichlet unit's theorem, and is called the *Log-unit lattice*.

We first remark that the group-ring $\mathbf{Z}[(\mathbf{Z}/f\mathbf{Z})^\times]$ acts on the group $(\mathbf{R}\otimes\mathbf{L})^\times$ in the following way: for any $g = \sum_\alpha g_\alpha \alpha \in \mathbf{Z}[(\mathbf{Z}/f\mathbf{Z})^\times]$ and $x \in (\mathbf{R}\otimes\mathbf{L})^\times$,

$$g\cdot x = \prod_{\alpha\in(\mathbf{Z}/f\mathbf{Z})^\times} \sigma_\alpha(x)^{g_\alpha},$$

where $\sigma_\alpha(\zeta_f) = \zeta_f^\alpha$. But σ_α acts as a permutation on the Archimedean embedding so that the embedding in the Log-unit lattice *commutes* with the action of $\mathbf{Z}[(\mathbf{Z}/f\mathbf{Z})^\times]$ in the following sense:

$$\mathrm{Log}(g\cdot x) = g\mathrm{Log}(x) \in \mathbf{R}[G],$$

for all $x \in (\mathbf{R}\otimes\mathbf{L})^\times$.

Henceforth, the cyclotomic units can be described using this action, as they correspond to the orbit of the element $\zeta_f - 1$ by the kernel, called the *augmentation ideal*, of $g \mapsto \sum_\alpha g_\alpha$:

$$\left\{ g\cdot(\zeta_f - 1) \mid \sum_\alpha g_\alpha = 0 \right\} \tag{5.3}$$

5.1.1.4. *An upper bound on the norm of* $Log(\zeta_f - 1)$. We also have that $\mathrm{Log}(\zeta_f - 1)$ is invertible, and with a small inverse so that we can compute efficiently. We first bound $\mathrm{Log}(\zeta_f - 1)$:

Lemma 5.1.4. *We have:*

$$\|Log(\zeta_f - 1)\|_\infty \leq \log f \qquad \textit{and} \qquad \|Log(\zeta_f - 1)\|_2 = O\Big(\sqrt{f}\Big).$$

Proof. The coordinates are given by

$$\mathrm{Log}(\zeta_f - 1)_\alpha = \mathrm{Log}(|\zeta_f^\alpha - 1|) = \log(|2\sin(\pi\alpha/f)|),$$

for any $\alpha \in (\mathbf{Z}/f\mathbf{Z})^\times$. Now, for $0 \leq x \leq \frac{1}{2}$ and $\alpha \in (\mathbf{Z}/f\mathbf{Z})^\times$, we have $\sin(\pi x) \geq 2x$ and we can consider that $0 \leq \frac{\alpha}{f} \leq \frac{1}{2}$. We deduce that $\|\mathrm{Log}(\zeta_f - 1)\|_\infty \leq \log\left(\frac{f}{4}\right)$ and

$$\|\mathrm{Log}(\zeta_f - 1)\|_2^2 \leq \sum_\alpha \log^2\left(\frac{f}{4\alpha}\right) \leq f\int_0^{\frac{1}{2}} \log^2\left(\frac{4}{x}\right)dx,$$

the latest integral being equal to $\frac{9}{2} + \frac{3}{\ln 2} + \frac{1}{\ln^2 2}$ entails the announced inequality. ∎

Remark. *The multiplication in the group ring $\mathbf{Z}[G]$ is quasi-linear as G is a finite abelian group. Indeed, we can use Fourier transform to reduce the multiplication to point-wise multiplications (see for instance [117]).*

5.1.1.5. *Fast rounding in the Log-unit lattice.* We can now describe the rounding algorithm, which essentially is a randomized coefficient-wise rounding using the orbital description of Equation 5.3.

Proof of Theorem 5.1.2. Without loss of generality, we can assume $N_{\mathbf{L}/\mathbf{Q}}(x) = 1$.

Then, using the description given by Equation 5.3 the problem is thus reduced to searching an unit u such that $\mathrm{Log}(u) \in \mathbf{Z}[G]$ which is close to $y = \frac{\mathrm{Log}(x)}{\mathrm{Log}(\zeta_f - 1)}$ and such that $\sum_\alpha \mathrm{Log}(u)_\alpha = 0$. The simplest idea consists in performing a coefficient wise rounding of the coefficients of the vector y. However this approach does not succeed all the time, but we can take advantage of the two possible choices in the rounding to closest integers to randomize the rounding—that is to say, by randomizing the choice of floor or ceil instead of relying deterministically to the round function $\lfloor \cdot \rceil$.

Formally, for $\alpha \neq 1$, we sample z_α following the unique distribution on the two elements set $\{\lfloor y_\alpha \rfloor, \lceil y_\alpha \rceil\}$ with expectation y_α. Then, z_1 is set at $-\sum_{\alpha \neq 1} z_\alpha$ to ensure $\sum_\alpha z_\alpha = 0$. Clearly, $u = z \cdot (\zeta_f - 1)$ verifies our requirements if

$$\left\| \mathrm{Log}(\zeta_f - 1)(y - z) \right\|_\infty = \mathrm{O}\left(\sqrt{f \log f}\right).$$

The Berry-Esseen theorem indicates that $|y_1 - z_1| \leq \sqrt{n}/\log n$ with probability $\Theta(1/\log n)$. The coordinates of

$$\mathrm{Log}(\zeta_f - 1)(y - z - (y - z)_1 \sigma_1)$$

are subgaussians of parameter $\|\mathrm{Log}(\zeta_f - 1)\|_2$. Therefore, using the estimation of Lemma 5.1.4, we know that their absolute values can all be bounded by $\mathrm{O}\left(\sqrt{f \log f}\right)$ except with probability at most $\Theta\left(\frac{1}{\log^2 f}\right)$. Hence, our requirement is fulfilled with probability $\Omega\left(\frac{1}{\log n}\right)$. We have $\mathrm{Log}(u) = z\mathrm{Log}(\zeta_f - 1)$ which can be computed in quasi linear time. Eventually a Fourier transform recovers $\sqrt{u\bar{u}}$, which is u up to an irrelevant torsion[1]. ∎

5.1.2 Extension to arbitrary cyclotomic fields

We now extend the result of Theorem 5.1.2 to arbitrary cyclotomic fields, that is proving:

1 One can compute u by simply removing the absolute values in the definition of Log, and taking any determination of complex logarithm. As we work inside a CM-field, this technicality is not needed.

Theorem 5.1.4. *Let* $\mathbf{L}$ *be the cyclotomic field of conductor* f*. There is a quasi-linear randomized algorithm that given any element in* $x \in (\mathbf{R} \otimes \mathbf{L})^\times$ *finds a unit* $u \in \mathscr{O}_\mathbf{L}^\times$ *such that for any field embedding* $\sigma : \mathbf{L} \to \mathbf{C}$ *we have*

$$\sigma\left(xu^{-1}\right) = 2^{O\left(\sqrt{f \log f}\right)} N_{\mathbf{L}/\mathbf{Q}}(x)^{\frac{1}{\varphi(f)}}.$$

In substance this proof is quite similar to the proof of the prime-power case but requires estimation of the evaluation of character's sum to bound the size of the roundings. As this computation is quite lengthy and not enlightening for the purposes of this chapter we defer it to Appendix 4.

5.2 REDUCTION OF ALGEBRAIC LATTICES IN CYCLOTOMIC FIELDS

With this balacing tool available we are now ready to introduce the recursion technique for reducing algebraic lattices.

Let h be a non-negative integer. In the following of this section we fix a tower of log-smooth conductor cyclotomic fields

$$\mathbf{L}_h^\uparrow = (\mathbf{Q} = \mathbf{L}_0 \subset \mathbf{L}_1 \subset \cdots \subset \mathbf{L}_h)$$

and denote by $1 = n_0 < n_1 < \cdots < n_h$ their respective degrees over $\mathbf{Q}$. Then we consider a free module $\mathcal{M}$ of rank d over the upper field $\mathbf{L}_h$, which is represented by a basis $(m_1, \ldots, m_d)$ given as the columns of a matrix $M \in \mathscr{O}_{\mathbf{L}_h}^{d \times d}$. For notational simplicity, in this section, we shall denote by $\langle a, b \rangle$ the $\mathscr{O}_\mathbf{L}$-module $a\mathscr{O}_\mathbf{L} \oplus b\mathscr{O}_\mathbf{L}$.

5.2.1 In-depth description of the algorithm

5.2.1.1. *Outer iteration.* In order to reduce the module $\mathcal{M}$ we adopt an iterative strategy to progressively modify the basis: for ρ steps a reduction pass over the current basis is performed, ρ being a parameter whose value is computed to optimize the complexity of the whole algorithm while still ensuring the reduceness of the basis; we defer the precise computation of this constant to Section 5.3. As in the LLL algorithm a size-reduction operation is conducted to control the size of the coefficients of the basis and ensure that the running time of the reduction is polynomial. Note that for number fields this subroutine needs to be adapted to deal with units of $\mathscr{O}_{\mathbf{L}_h}$ when rounding. The specificities of this size-reduction are the matter of Paragraph 5.2.1.5.

5.2.1.2. *Step reduction subroutine.* We now take a look at the step reduction pass, once the size-reduction has occurred. As observed in Chapter 2, the textbook LLL algorithm is build on a simple idea: make the reduction process boiling down to the treatment of rank two modules and more precisely to iteratively reduce *orthogonally projected* rank two submodules. We are using the same paradigm here and this step reduction pass over the current basis is a sequence of reduction of projected rank 2 $\mathscr{O}_{\mathbf{L}_h}$−modules. However

on the contrary to the LLL algorithm we do not proceed progressively along the basis, but reduce $\lfloor d/2 \rfloor$ independent rank 2 modules at each step. This design enables an efficient parallel implementation which reduces submodules simultaneously, in the same way that the classical LLL algorithm can be parallelized [77, 161].

Formally, given the basis of $\mathcal{M}$ collected in the matrix M, let us denote by r_j the vector $\left(R_{j,j}, R_{j+1,j} = 0\right)$, and r'_j the vector $\left(R_{j+1,j}, R_{j+1,j+1}\right)$ where R is the R-part of the QR-decomposition of M. The module $\mathcal{R}_i$ encodes exactly the projection of $\mathcal{M}_i = \langle m_{i-1}, m_i \rangle$ over the orthogonal space to the first $i-1$ vectors $(m_1, \ldots, m_{i-1})$. In order to recursively call the reduction algorithm on $\mathcal{R}_i$ we need to *descend* it to the subfield $\mathbf{L}_{h-1}$.

5.2.1.3. *Interlude: descending to cyclotomic subfields.* Remark now that since $\mathbf{L}_h$ is a cyclotomic extension of the cyclotomic field $\mathbf{L}_{h-1}$, there exists a root of unity ξ such that

$$\mathscr{O}_{\mathbf{L}_h} = \mathscr{O}_{\mathbf{L}_{h-1}} \oplus \xi \mathscr{O}_{\mathbf{L}_{h-1}} \oplus \cdots \oplus \xi^{q_h - 1} \mathscr{O}_{\mathbf{L}_{h-1}}.$$

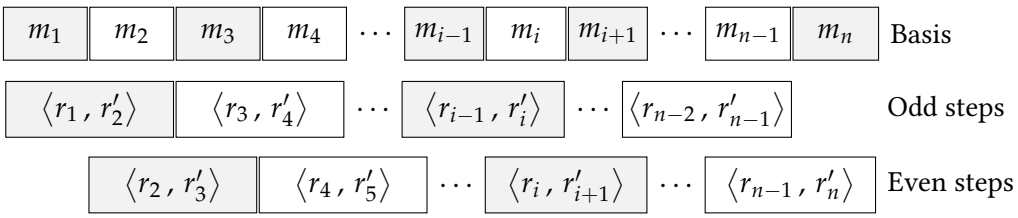
for $q_h = n_h / n_{h-1}$ being the relative degree of $\mathbf{L}_h$ over $\mathbf{L}_{h-1}$. As a consequence, the module $\mathcal{R}_i$ decomposes over $\mathscr{O}_{\mathbf{L}_{h-1}}$ as:

$$\begin{aligned}
\mathcal{R}_i &= r_i \mathscr{O}_{\mathbf{L}_h} \oplus r'_{i+1} \mathscr{O}_{\mathbf{L}_h} \\
&= r_i \mathscr{O}_{\mathbf{L}_{h-1}} \oplus \xi r_i \mathscr{O}_{\mathbf{L}_{h-1}} \oplus \cdots \oplus \xi^{q_h - 1} r_i \mathscr{O}_{\mathbf{L}_{h-1}} \oplus \\
&\quad r'_{i+1} \mathscr{O}_{\mathbf{L}_{h-1}} \oplus \xi r'_{i+1} \mathscr{O}_{\mathbf{L}_{h-1}} \oplus \cdots \oplus \xi^{q_h - 1} r'_{i+1} \mathscr{O}_{\mathbf{L}_{h-1}},
\end{aligned}$$

yielding a basis of $\mathcal{R}_i$ viewed as a free $\mathscr{O}_{\mathbf{L}_{h-1}}$-module of rank $2 \times q_h$. This module can then recursively reduced, this time over a tower of height $h-1$. This conversion from an $\mathscr{O}_{\mathbf{L}_h}$-module to an $\mathscr{O}_{\mathbf{L}_{h-1}}$ module is referred as the function **Descend**. Conversely, any vector $u \in \mathscr{O}_{\mathbf{L}_{h-1}}^{2q_h}$ can be seen with this decomposition as a vector of $\mathscr{O}_{\mathbf{L}_h}^2$ by grouping the coefficients as $\left(\sum_{i=1}^{q_h} u[i]\xi^i, \sum_{i=1}^{q_h} u[q_h + 1 + i]\xi^i\right)$. We denote by **Ascend** this conversion.

5.2.1.4. *Back on the step reduction.* As mentioned in Paragraph 5.2.1.2, we start by reducing—with a recursive call after descending—all the modules $\mathcal{R}_{2i} = \langle r_{2i-1}, r'_{2i} \rangle$ for $1 \leq i \leq \lfloor d/2 \rfloor$, so that each of these reductions yields a small element of the submodule $\mathcal{M}_{2i} = \langle m_{2i-1}, m_{2i} \rangle$; which is then *completed*[2] in a basis of $\mathcal{M}_{2i}$. But on the contrary of the classical LLL reduction, this sequence of pairwise independent reductions does not make interact the elements m_{2i} and m_{2i+1}, in the sense that no reduction of the module projected from $\langle m_{2i}, m_{2i+1} \rangle$ is performed. To do so, we then perform the same sequence of pairwise reductions but with all indices shifted by 1: we reduce the planes $\langle r_{2i}, r'_{2i+1} \rangle$ for each $1 \leq i \leq \lfloor d/2 \rfloor$, as depicted in the following diagram:

2 The precise definition of this completion and lifting is given in Paragraph 5.2.1.7.



5.2.1.5. *Unit-size-reduction for $\mathcal{O}_{\mathbf{L}_h}$-modules.* To adapt the size-reduction process to the module setting, one needs to adjust the rounding function. When $\mathbf{L}_h = \mathbf{Q}$, the rounding boils down to finding the closest element in $\mathcal{O}_{\mathbf{L}} = \mathbf{Z}$, which is encompassed by the round function $\lceil \cdot \rfloor$. In the higher-dimensional context, we need to approximate any element of $\mathbf{L}_h$ by *a* close element of $\mathcal{O}_{\mathbf{L}_h}$.

Note that finding *the* closest integral element is not efficiently doable. The naive approach to this problem consists in reducing the problem to the resolution of the closest integer problem in the Euclidean lattice of rank n_h given by $\mathcal{O}_{\mathbf{L}_h}$ under the Archimedean embedding. However, up to our knowledge, no exponential speedup exists using its particular structure compared to sieving or enumeration in this lattice.

Nonetheless, finding a target *close enough* to the target suffices for our application. As such we simply define the rounding of an element $\alpha \in \mathbf{L}_h$ as the integral rounding on each of its coefficients when represented in the power base of $\mathbf{L}_h$.

We add here an important and necessary modification: before the actual size-reduction occurred, we compute a unit u using Theorem 5.1.1 close to $R_{i,i}$. This routine is denoted by **Unit**. The vector M_i is then divided by u. While not changing the algebraic norms of the elements, this technicality forces the Archimedean embeddings of the coefficients to be balanced and helps the reduced matrix to be well-conditioned. This avoids a blow-up of the precision required during the computation. This modified size-reduction is fully described in Algorithm 20, **Size-Reduce**.

Algorithm 20 — Size-Reduce

Input : R-factor of the QR-decomposition of $M \in \mathcal{O}_{\mathbf{L}_h}^{d \times d}$
Output : A unimodular transformation U representing the size-reduced basis obtained from M.

$U \leftarrow \mathrm{Id}_{d,d}$
for $i = 1$ **to** d **do**
  $D \leftarrow D_i(\mathbf{Unit}(R_{i,i}))$ // D_i is a dilation matrix
  $(U, R) \leftarrow (U, R) \cdot D^{-1}$
  for $j = i - 1$ **downto** 1 **do**
    $\sum_{\ell=0}^{n-1} r_\ell X^\ell \leftarrow R_{i,j}$ // Extraction as a polynomial
    $\mu \leftarrow \sum_{\ell=0}^{n-1} \lfloor r_\ell \rceil X^\ell$ // Approximate rounding of $R_{i,j}$ in $\mathcal{O}_{\mathbf{L}_h}$
    $(U, R) \leftarrow (U, R) \cdot T_{i,j}(-\mu)$ // $T_{i,j}$ is a shear matrix
  end for
end for
return U

5.2.1.6. *Reduction of the leaves.* As the recursive calls descend along the tower of number fields, the bottom of the recursion tree requires reducing $\mathcal{O}_{\mathbf{L}_0} (= \mathcal{O}_{\mathbf{Q}} = \mathbf{Z})$-modules, that is Euclidean lattices. As a consequence, the step reduction performs calls to a reduction oracle for plane Euclidean lattices. For the sake of efficiency we adapt Schönhage's algorithm [148] to reduce these lattices, which is faster than the traditional Gauss' reduction. This algorithm is an extension to the bidimensional case of the half-GCD algorithm, in the same way, that Gauss' algorithm can be seen as a bidimensional generalization of the classical GCD computation.

The original algorithm of Schönhage only deals with the reduction of binary quadratic forms, but can be straightforwardly adapted to reduce rank 2 Euclidean lattices, and to return the corresponding unimodular transformation matrix. In all of the following, we denote by **Schonhage** this modified procedure.

5.2.1.7. *The lifting phase.* As explained in Paragraph 5.2.1.2, we recursively call the reduction procedure to reduce the descent of projected modules of rank 2 of the form $\mathcal{R}_i = \langle r_i, r'_{i+1} \rangle$, over $\mathbf{L}_{h-1}$, yielding a unimodular transformation $U' \in \mathcal{O}_{\mathbf{L}_{h-1}}^{2q_h \times 2q_h}$ where q_h is the relative degree of $\mathbf{L}_h$ over $\mathbf{L}_{h-1}$.

From U', we can find random short elements in the module by computing a small linear combination of the first columns. Applying **Ascend**, we deduce some short $x = m_i a + m_{i+1} b$. But then to replace m_i by x in the current basis, we need to complete this vector into a basis (x, y) of $\mathcal{M}_i$ over $\mathcal{O}_{\mathbf{L}_h}$. Doing so boils down to complete a vector of $\mathcal{O}_{\mathbf{L}_h}^2$ into a unimodular transformation. Indeed, suppose that such a vector y is found and denote by

(a, b) and (v, u) the respective coordinates of x and y in the basis (m_i, m_{i+1}). By preservation of the volume we have without loss of generality:

$$1 = \det \begin{pmatrix} a & v \\ b & u \end{pmatrix} = au - bv.$$

Therefore finding the element y to complete x reduces to solving the Bézout equation in the unknown u and v

$$au - bv = 1 \tag{5.4}$$

over the ring $\mathcal{O}_{\mathbf{L}_h}$. Since this ring is in general not Euclidean we can not apply directly the Euclidean algorithm to solve this equation as an instance of the extended GCD problem. However, we can use the algebraic structure of the tower $\mathbf{L}_h^{\uparrow}$ to recursively reduce the problem to the rational integers. This *generalized* Euclidean algorithm works as follows:

If $\mathbf{L}_h = \mathbf{Q}$: then the problem is an instance of extended GCD search, which can be solved efficiently by the binary-GCD algorithm.

If the tower $\mathbf{L}_h^{\uparrow}$ is not trivial: we make use of the structure of $\mathbf{L}_h^{\uparrow}$ and first descend the problem to the subfield $\mathbf{L}_{h-1}$ by computing the relative norm $N_{\mathbf{L}_h/\mathbf{L}_{h-1}}$ of the elements a and b; then by recursively calling the algorithm on these elements $N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(a)$ and $N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(b)$, we get two algebraic integers μ and ν of $\mathcal{O}_{\mathbf{L}_{h-1}}$ fulfilling the equation:

$$\mu N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(a) - \nu N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(b) = 1. \tag{5.5}$$

But then remark that for any element $\alpha \in \mathcal{O}_{\mathbf{L}_h}$ we have, using the comatrix formula and the definition of the norm as a determinant that: $N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(\alpha) \in \alpha\mathcal{O}_{\mathbf{L}_h}$, so that $\alpha^{-1} N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(\alpha) \in \mathcal{O}_{\mathbf{L}_h}$. Then, from Equation 5.5:

$$a \cdot \underbrace{\mu\, a^{-1} N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(a)}_{:= u \in \mathcal{O}_{\mathbf{L}_h}} - b \cdot \underbrace{\nu\, b^{-1} N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(b)}_{:= v \in \mathcal{O}_{\mathbf{L}_h}} = 1,$$

as desired.

Reduction of the size of solutions: The elements u, v found by the algorithm are not necessarily the smallest possible elements satisfying Equation 5.4. To avoid a blow-up in the size of the coefficients lifted, we do need to control the size of the solution at each step. Since the function **Size-Reduce** preserves the determinant by construction and reduces the norm of the coefficients, we can use it to reduce the bitsize of u, v to (roughly) the bitsize of a and b.

The translation of this method in pseudocode is given in Algorithm 21, **G-Euclide**.

Algorithm 21 — G-Euclide, Lift

Function ***G-Euclide***:
  Input : Tower of number fields $\mathbf{L}_h^{\uparrow}$, $a, b \in \mathbf{L}_h$.
  Output : $u, v \in \mathbf{L}_h$, such that $au + bv = 1$
  if $\mathbf{L}_h = \mathbf{Q}$ **then return** $\mathbf{ExGcd}(a, b)$
  $\mu, \nu \leftarrow \mathbf{G\text{-}Euclide}\left(\mathbf{L}_{h-1}^{\uparrow}, N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(a), N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(b)\right)$
  $\mu', \nu' \leftarrow \mu\, a^{-1} N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(a), \nu\, b^{-1} N_{\mathbf{L}_h/\mathbf{L}_{h-1}}(b)$
  $W \leftarrow \begin{pmatrix} a & \nu' \\ b & \mu' \end{pmatrix}$
  $V \leftarrow \mathbf{Size\text{-}Reduce}(\mathbf{Orthogonalize}(W))$
  return $W \cdot V[2]$

Function ***Lift***:
  Input : Tower of number fields $\mathbf{L}_h^{\uparrow}$, unimodular matrix $U' \in \mathcal{O}_{\mathbf{L}_{h-1}}^{2q_h}$
  Output : Unimodular matrix $U \in \mathcal{O}_{\mathbf{L}_h}^{2\times 2}$
  $a, b \leftarrow \mathbf{Ascend}(\mathbf{L}_h, U[1])$
  $\mu, \nu \leftarrow \mathbf{G\text{-}Euclide}\left(\mathbf{L}_{h-1}^{\uparrow}, a, b\right)$
  $U \leftarrow \begin{pmatrix} a & \nu \\ b & \mu \end{pmatrix}$
  return U

The number of bits needed to represent the relative norms does not depend on the subfield, and the size-reduction forces the output vector to have the same bitsize as the input one. This remark is the crux of the quasilinearity of the **G-Euclide**, as stated in Lemma 5.3.3.

Remark that the algorithm needs $N_{\mathbf{L}_h/\mathbf{Q}}(a)$ to be prime with $N_{\mathbf{L}_h/\mathbf{Q}}(b)$. We assume that we can always find quickly such a, b with a short x. This will lead to Heuristic 5.3.1, and the validity of this assumption is discussed in Section 5.5.3.

5.2.2 Wrapping-up

The full outline of the reduction if given in Algorithm 22 and a schematic overview of the recursive steps is provided in the diagram of Figure 1.

Algorithm 22 — Reduce

Input : Tower of cyclotomic fields $\mathbf{L}_h^{\uparrow}$, Basis $M \in \mathcal{O}_{\mathbf{L}_h}^{d \times d}$ of the $\mathcal{O}_{\mathbf{L}_h}$−module $\mathcal{M}$

Output : A unimodular transformation $U \in \mathcal{O}_{\mathbf{L}_h}^{d \times d}$ representing a reduced basis of $\mathcal{M}$.

if $d = 2$ **and** $\mathbf{L}_h = \mathbf{Q}$ **then return** **Schonhage**(M)
for $i = 1$ **to** ρ **do**
  $R \leftarrow$ **Orthogonalize**(M)
  $U_i \leftarrow$ **Size-Reduce**(R)
  $(M, R) \leftarrow (M, R) \cdot U_i$
  for $j = 1 + (i \mod 2)$ **to** d **by step of** 2 **do**
    $M' \leftarrow$ **Descend**$(\mathbf{L}_{h-1}^{\uparrow}, R[j : j+1, j : j+1])$
    $U' \leftarrow$ **Reduce**$(\mathbf{L}_{h-1}^{\uparrow}, M')$
    $(U_i, M) \leftarrow (U_i, M) \cdot$ **Lift**(U')
  end for
end for
return $\prod_{i=1}^{\rho} U_i$

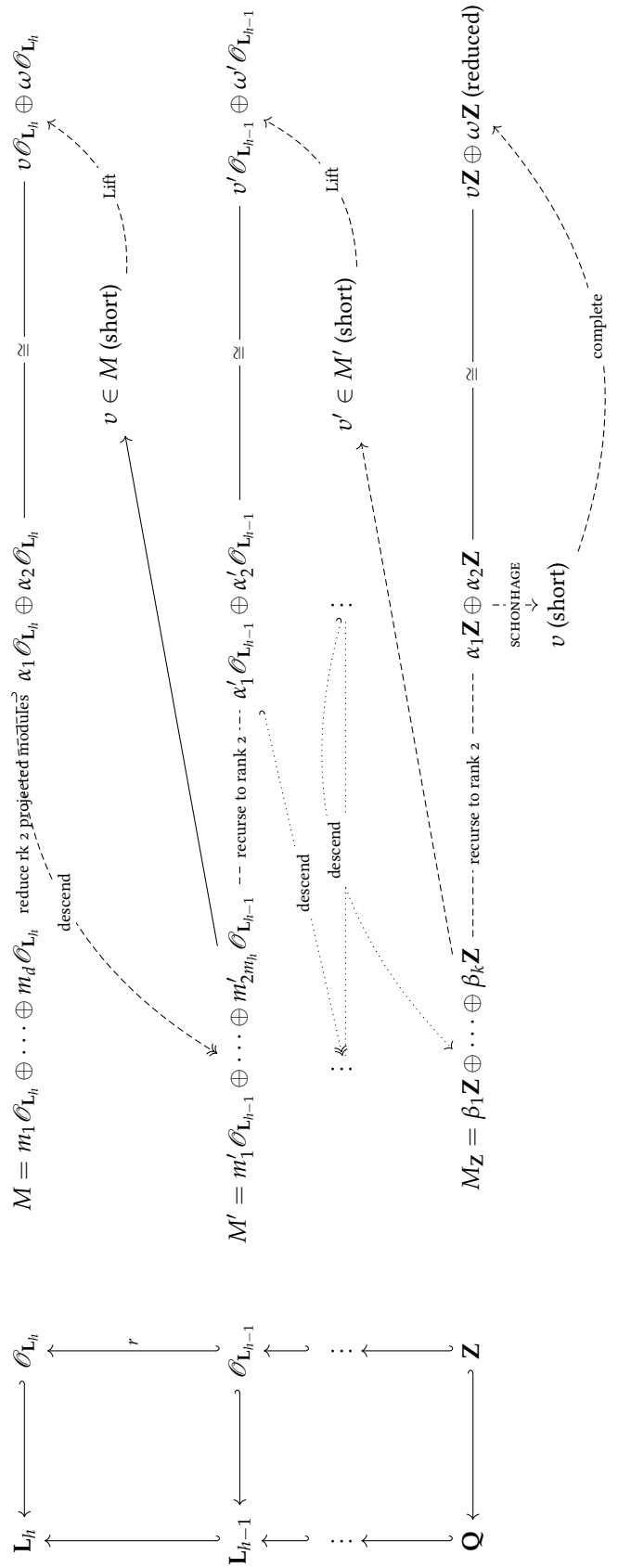


Figure 1: Schematic view of the recursive call of reductions.

5.3 COMPLEXITY ANALYSIS

In this section, we devise the complexity of the Algorithm 22 and of its approximation factor. More formally we prove the following theorem:

Theorem 5.3.1. *Let f be a log-smooth integer. The complexity of the algorithm* **Reduce** *on rank two modules over $\mathbf{K} = \mathbf{Q}[x]/\Phi_f(x)$, represented as a matrix M whose number of bits in the input coefficients is uniformly bounded by $B > n$, is heuristically a $\tilde{O}(n^2 B)$ with $n = \varphi(f)$. The first column of the reduced matrix has its coefficients uniformly bounded by $2^{\tilde{O}(n)} (\operatorname{covol} M)^{\frac{1}{2n}}$.*

5.3.1 Setting

Let $h > 0$ be a non-negative integer. In the following of this section we fix a tower of cyclotomic fields $\mathbf{K}_h^{\uparrow} = (\mathbf{Q} = \mathbf{K}_0 \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h)$ with log-smooth conductors and denote by $1 = n_0 < n_1 < \cdots < n_h$ their respective degrees over $\mathbf{Q}$. We consider a free module $\mathcal{M}$ of rank d over the upper field $\mathbf{K}_h$, given by one of its basis, which is represented as a matrix $M \in \mathcal{O}_{\mathbf{K}_h}^{d \times d}$. In all of the following, for any matrix A with coefficients in $\mathbf{K}_h$ we denote by $\|A\|$ the 2-norm for matrices.

We aim at studying the behavior of the reduction process given in Algorithm 22 on the module $\mathcal{M}$; as such we denote generically by $X^{(\tau)}$ the value taken by any variable X appearing in the algorithm at *the end* of step $i = \tau$, for $0 \leq \tau \leq \rho$. For instance $R^{(0)}$ denotes the R-part of the orthogonalization of M and $M^{(\rho)}$ represents the reduced basis at the end of the algorithm.

Since the implementation of the algorithm is done using floating-point arithmetic, we need to set a precision which is sufficient to handle the internal values during the computation. To do so we set:

$$p = \log \frac{\max_{\sigma : \mathbf{K}_h \to \mathbf{C}, R_{i,i} \in R} \sigma(R_{i,i})}{\min_{\sigma : \mathbf{K}_h \to \mathbf{C}, R_{i,i} \in R} \sigma(R_{i,i})},$$

where the σ runs over the possible field embeddings and the $R_{i,i}$ are the diagonal values of the R part of the QR-decomposition of the input matrix of the reduction procedure. We will prove as a byproduct of the complexity analysis that taking a precision of $O(p)$ suffices.

For technical reasons which will appear in the subsequent proofs, we introduce a constant $\alpha > 0$ which will be optimized at the end of our analysis. It essentially encodes the approximation factor of the reduction. Eventually, we set the variable ε to be equal to $1/2$. This apparently odd choice allows us to state our theorems with sufficient generality to reuse them in the enhanced proof of the reduction algorithm with symplectic symmetries, as detailed in Section 5.4, with a different value.

The whole set of notations used in the analysis is recalled in Table 5.1.

h	Height of the tower
n_h	Absolute height $[\mathbf{K}_h : \mathbf{Q}]$
p	bound on the precision used by the reduction
ε	$1/2$
i	Current outmost loop number ($1 \leq i \leq \rho$) iteration
α	Constant to be optimized

Table 5.1: Notations used in the complexity analysis. p is of course set to be larger to the bitsize of the input matrix.

5.3.2 Overview of the proof

Before going into the details of the proof, we lay its blueprint. We start by estimating the approximation factor of the reduction and deduce a bound in $\mathrm{O}(d^2 \log p)$ on the number of rounds ρ required to achieve the reduction the module $\mathcal{M}$, where p is the precision needed to handle the full computation. We then prove that the limiting factor for the precision is to be sufficiently large to represent the shortest Archimedean embedding of the norm of the Gram-Schmidt orthogonalization of the initial basis. We then devise a bound by looking at the sum of all the bit sizes used in the recursive calls and concludes on the complexity. The critical part of the proof is to use the potential to show that dividing the degrees by $\frac{d}{2}$ leads to a multiplication by a factor at most in $\mathrm{O}(d^2)$ of the sum of all the precisions in the recursive calls, instead of the obvious $\mathrm{O}(d^3 \log p)$.

5.3.3 A bound on the number of rounds and the approximation factor of the reduction

We define here a set of tools to study the approximation factor of the reduction, by approximating it by an iterative linear operator on the family of volumes of the submodules $\mathcal{M}_i = m_1\mathbf{Z} \oplus \cdots \oplus m_i\mathbf{Z}$ for $1 \leq i \leq d$. This method is quite similar to the one used by Hanrot *et al.* in [74] to analyze the BKZ algorithm by studying a dynamical system.

To ease the computation of the number of rounds, we can without loss of generality, scale the input matrix and suppose that:

$$\operatorname{covol} \mathcal{M} = |N_{\mathbf{K}_h/\mathbf{Q}}(\det M)|^{\frac{1}{d}} = 2^{-(d+1)(1+\varepsilon)\alpha n_h^2}.$$

We only do so for this subsection.

5.3.3.1. *Potential and volumes of flags.* A *global* measure of reduceness of a Euclidean lattice is its potential. An $\mathcal{O}_{\mathbf{K}_h}$-analog of this constant can be defined in a similar manner by using the algebraic norm to replace the Euclidean norm over $\mathbf{R}^n$.

Definition 5.3.1 (Potential). *Let $(m_1, \ldots, m_d)$ be a basis of the module $\mathcal{M}$ given as the columns of a matrix $M \in \mathcal{O}_{\mathbf{K}_h}^{d\times d}$, and let R be the R-part of its QR-decomposition. Its* log-potential *is defined as:*

$$\Pi(M) = \sum_{i=1}^{d} \log \operatorname{covol} \mathcal{M}_i = \sum_{i=1}^{d} (d-i) \log N_{\mathbf{K}_h/\mathbf{Q}}(R_{i,i}).$$

As in the Euclidean case, a *local* tool to analyze the evolution of a basis $(m_1, \ldots, m_d)$ of a lattice Λ, through a reduction, is the *profile* of the volumes associated with the flag of a basis, namely the family:

$$\operatorname{covol}(\mathcal{M}_1), \ldots, \operatorname{covol}(\mathcal{M}_i), \ldots, \operatorname{covol} \Lambda.$$

As for the potential, we define the profile of the flag in a similar way with the algebraic norm on $\mathbf{K}_h$, but for technical reasons, we quadratically twist it with the constant $\alpha > 0$.

Definition 5.3.2 (Flag profile). *Let $(m_1, \ldots, m_d)$ be a basis of the module $\mathcal{M}$ given as the columns of a matrix $M \in \mathcal{O}_{\mathbf{K}_h}^{d\times d}$, and let R be the R-part of its QR-decomposition. Its* profile *is the vector $\mu(M) \in \mathbf{R}^d$ defined by:*

$$\mu(M)_j = \sum_{k=1}^{j} \left(\log N_{\mathbf{K}_h/\mathbf{Q}}(R_{k,k}) + 2k(1+\varepsilon)\alpha n_h^2\right), \qquad \text{for } 1 \le j \le d.$$

The following lemma gives an estimate of the norm of the profile in terms of the parameters of the algorithm and of the input bitsize.

Lemma 5.3.1. *With the same notations as in Definition 5.3.2, we have:*

$$\|\mu(M)\|_2 \le (2+\varepsilon)\alpha d^2 n_h p$$

Proof. We have $|N_{\mathbf{K}/\mathbf{Q}}(\det(M))| \le 1$ so for each i, and each embedding σ, we have that $|\sigma(R_{i,i})| \le 2^p$. Now we compute:

$$\begin{aligned}\frac{\|\mu(M)\|_2^2}{d} &\le \max_{j=1,\ldots,d-1} \left\{ \sum_{k\le j} \left(\log N_{\mathbf{K}_h/\mathbf{Q}}(R_{k,k}) + 2k(1+\varepsilon)\alpha n_h^2\right) \right\} \\ &\le d n_h p + d(d-1)(1+\varepsilon)\alpha n_h^2\end{aligned}$$

which implies the result. ∎

5.3.3.2. *A family of step operators.* To study the reduction steps, we define the following linear operators

$$\delta_j : \left| \begin{array}{ccc} \mathbf{R}^d & \longrightarrow & \mathbf{R}^d \\ v & \longmapsto & (w_\ell)_\ell = \begin{cases} \frac{v_{j-1}+v_{j+1}}{2} & \text{if } \ell = j \\ v_j & \text{if } \ell = j+1 \\ v_\ell & \text{else} \end{cases} \end{array} \right. , \tag{5.6}$$

for each $1 \leq j \leq d$. These operators provide an upper bound on the profile of a basis after a reduction at index j. To encode the behavior of a full round of reduction we define the operators:

$$\Delta_o = \prod_{i=1 \,|\, i \text{ odd}} \delta_i, \quad \text{and} \quad \Delta_e = \prod_{i=2 \,|\, i \text{ even}} \delta_i,$$

to define inductively the sequence:

$$\begin{aligned} \mu^{(1)} &= \mu(M^{(1)}) \\ \mu^{(i)} &= \Delta_o\left(\mu^{(i-1)}\right) \quad \text{if } i = 0 \pmod 2 \quad \text{else} \quad \Delta_e\left(\mu^{(i-1)}\right) \end{aligned}$$

Remark. *By the constraint on the volume of $\mathcal{M}$ to be equal to $2^{-d(d+1)(1+\varepsilon)\alpha n_h^2}$, we have for all $1 \leq i \leq \rho$, that $\mu_d^{(i)} = 0$.*

Proposition 5.3.1 (Exponential decay of $\|\mu^{(i)}\|_2$)**.** *For all odd i, we have,*

$$\left|\mu_1^{(i)}\right| \leq \mathrm{e}^{-\frac{\pi^2(i-1)}{2d^2}} \|\mu^{(1)}\|_2$$

and

$$\|\mu^{(i+1)}\|_2 \leq 2\mathrm{e}^{-\frac{\pi^2(i-1)}{2d^2}} \|\mu^{(1)}\|_2.$$

Proof. Note that $\Delta_o \circ \Delta_e$ depends only on the odd coordinates, so let Δ be its restriction to them in the domain and codomain. Remark that for all $1 \leq k \leq \lceil \frac{d-1}{2} \rceil$ the vector

$$\left(\sin\left(\frac{jk\pi}{2\lfloor d/2 \rfloor}\right)\right)_j$$

is an eigenvector of Δ of associated eigenvalue $\cos\left(\frac{k\pi}{2\lfloor d/2 \rfloor}\right)^2$. A direct computation ensures that the eigenvectors are orthogonal. Since $2\lfloor d/2 \rfloor \leq d$, we use the trivial bound $\left|\cos\left(\frac{k\pi}{d}\right)\right| \leq \cos\left(\frac{\pi}{d}\right)$ in addition to the convexity bound

$$\ln(\cos(\pi/d)) < -\frac{\pi^2}{2d^2}$$

to obtain:

$$\sum_{k=1 \text{ odd}} \left(\mu^{(i)}\right)_k^2 \leq \mathrm{e}^{-\frac{\pi^2(i-1)}{2d^2}} \|\mu^{(1)}\|_2^2.$$

This implies the first statement and

$$\sum_{k=2 \text{ even}} \left(\mu^{(i+1)}\right)_k^2 \leq \sum_{k=1 \text{ odd}} \left(\mu^{(i)}\right)_k^2$$

implies the second. ∎

Remark (A "physical" interpretation of Δ)**.** *The operator Δ introduced in the proof of Proposition 5.3.1 acts as a* discretized *Laplacian operator on the discrete space indexed by $\{1, \ldots, d\}$, for a metric where two consecutive integers are at distance 1. Then, the action of Δ through the iterations $1 \leq i \leq \rho$ are reminiscent of the diffusion property of the solution of the* heat equation *($\frac{\partial u}{\partial t} = \alpha \Delta u$), whose characteristic time is quadratic in the diameter of the space.*

5.3.3.3. *A computational heuristic.* We now relate the behavior of the sequences of μ to the values taken by $R^{(i)}$. In order to do so, we introduce a computational heuristic on the behavior of the **Lift**function, asserting that the lifting phase does not blow up the size of the reduced vectors.

Heuristic 5.3.1 (Size of lifting). *For any $1 \leq i \leq \rho$ and any $1 \leq j \leq d$ where a call to **Lift** happened:*

$$N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i+1)}\right) \leq \min\left(2^{\alpha n_h^2}\sqrt{N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right).$$

A discussion on the validity of this heuristic is done in Section 5.5.3. However, we *do not* perform a local reduction if the following condition is fulfilled, up to the approximation error due to the representation at finite precision[3]

$$N_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j}^{(i)}) \leq \min\left(2^{(1+\varepsilon)\alpha n_h^2}\sqrt{N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right).$$

From Heuristic 5.3.1 we can show by a direct induction on i that the sequence of $\mu^{(i)}$ is an over-approximation of the flag profile at step i. More precisely we have:

Lemma 5.3.2. *Under Heuristic 5.3.1, for any $1 \leq i \leq \rho$:*

$$\mu\left(M^{(i)}\right) \leq \mu^{(i)},$$

where the comparison on vectors is taken coefficient-wise.

5.3.3.4. *A bound on the approximation factor and number of rounds.* We can now conclude this paragraph by giving a quasiquadratic bound on the number of rounds:

Theorem 5.3.2. *Assuming that ρ is even and $\rho > \frac{2d^2}{\pi^2}\ln((2+\epsilon)\alpha d^2 n_h p)$, we have that*

$$N_{\mathbf{K}_h/\mathbf{Q}}(R_{1,1}^{(\rho)}) \leq 2^{(d-1)(1+\varepsilon)\alpha n_h^2+1}|N_{\mathbf{K}_h/\mathbf{Q}}(\det M)|^{\frac{1}{d}}.$$

3 More precisely, if the precision used when performing this testing is p, then if we are certain that

$$N_{\mathbf{K}_h/\mathbf{Q}}(R_{i,i}) \leq \min\left(2^{(1+\varepsilon)\alpha n_h^2}\sqrt{N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right),$$

no local reduction is called, else we have

$$N_{\mathbf{K}_h/\mathbf{Q}}(R_{i,i}) \geq \min\left(2^{(1+\varepsilon)\alpha n_h^2}\sqrt{N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right)(1-2^{-\Omega(p)})$$

and a recursive local reduction is called, the multiplicative error term coming from the approximation error committed by the approximation of the values $R_{*,*}$ at precision p.

Proof. By taking the exponential of both sides of the inequality of Lemma 5.3.2, we have:

$$N_{\mathbf{K}_h/\mathbf{Q}}(R_{1,1}^{(\rho+1)}) \leq 2^{\mu_1^{(\rho+1)}-2(1+\varepsilon)\alpha}.$$

Recall that we forced $\left|N_{\mathbf{K}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{d}} = 2^{-(d+1)\alpha n_h^2(1+\varepsilon)}$, so that:

$$N_{\mathbf{K}_h/\mathbf{Q}}(R_{1,1}^{(\rho+1)}) \leq 2^{(d-1)(1+\varepsilon)\alpha n_h^2+\mu_1^{(\rho+1)}} \left|N_{\mathbf{K}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{d}}.$$

By Proposition 5.3.1, we know that $\mu_1^{(\rho+1)} \leq e^{-\frac{\pi^2\rho}{2d^2}} \|\mu^{(1)}\|_2$. Since we have:

$$\begin{aligned}\ln|\mu_1^{(\rho+1)}| &\leq \ln\|\mu^{(1)}\|_2 - \frac{\rho\pi^2}{2d^2} \\ &\leq \ln((2+\epsilon)\alpha d^2 n_h p) - \frac{\rho\pi^2}{2d^2} \leq \quad 0,\end{aligned}$$

using Lemma 5.3.1 and the hypothesis on ρ together with the fact that $d > 1$. All in all $|\mu_1^{(\rho)}| \leq 1$ and which entails the desired inequality. ∎

With mild assumptions on the relative size of the parameters α, n_h, d and p we have the following rewriting of Theorem 5.3.2.

Corollary 5.3.1. *Suppose that $\alpha = \log^{O(1)}(n_h)$ and that $p > n_h + d$, then taking $\rho = O(d^2 \log(p))$ is sufficient to reduce the module $\mathcal{M}$ and such that the algebraic norm of the first vector is bounded by a*

$$2^{\tilde{O}(dn_h^2)} |N_{\mathbf{K}_h/\mathbf{Q}}(\det M)|^{\frac{1}{d}}.$$

Remark. *If the caller makes a similar heuristic with a α', then we need $\alpha' > \alpha \cdot 2(1+\epsilon)\frac{d-1}{d}$ and any such value is plausible.*

5.3.4 Time complexity of the toplevel reduction

Now that we have an estimate of the number of rounds, we can aim at bounding the complexity of each round, without counting the recursive calls, in a first time. To do so we will look independently at each of the part of a round, namely at the complexity of **Orthogonalize**, **Reduce**and **Lift**. Since the lifting algorithm performs a size-reduction, we first give a fine-grained look at the **Size-Reduce**function.

5.3.4.1. *Complexity and quality of **Size-Reduce**.* The quantitative behavior of the **Size-Reduce** procedure is encoded by the following theorem, given in all generality for arbitrary matrices over a cyclotomic field.

Theorem 5.3.3. *Let A be a matrix of dimension d whose coefficients lie in the cyclotomic field $\mathbf{K} = \mathbf{Q}[\zeta_f]$, and $n = \varphi(f)$. We are given a non-negative integer $p > 0$, where $\|A\|, \|A^{-1}\| \leq 2^p$ and such that $\sqrt{n \log n \log\log n} + d \log n < p$. By calling the algorithm **Orthogonalize** and **Size-Reduce**, we can find in time*

$$O\left(d^2 np\left(1 + \frac{d}{\log p}\right)\right)$$

an integral triangular matrix $U \in \left(\mathcal{O}_{\mathbf{K}}^{\times}\right)^{n\times n}$, *such that* $\|U\| \leq 2^{O(p)}$, *and a matrix* $R + E$, *such that* $\|E\| \leq 2^{-p}$, *with* R *being the* R*-factor of the* QR *decomposition of* AU *and*

$$\kappa(AU) \leq \left(\frac{\max_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)^{\frac{1}{n}} 2^{O\left(\sqrt{n \log n \log\log n} + d \log n\right)},$$

for $\kappa(X) = \|X\|\|X^{-1}\|$ *being the condition number of* X.

Proof. See Appendix 3. ∎

Corollary 5.3.2. *Suppose that:*

$$\|M^{(0)}\|, \|{M^{(0)}}^{-1}\| \leq 2^p \qquad \text{and} \qquad d \log n_h + \sqrt{n_h \log n_h \log\log n_h} < p.$$

Then, we have the following bound on the condition number of $M^{(i)}$, *valid for any loop index* $1 \leq i \leq \rho$*:*

$$\kappa\left(M^{(i)}\right) \leq 2^{2p + O\left(\sqrt{n_h \log n_h \log\log n_h} + d \log n_h\right)},$$

and the call of the procedure **Size-Reduce** *at this* i*-th round has complexity*

$$O\left(d^2 n_h p \left(1 + \frac{d}{\log p}\right)\right)$$

and requires a $O(p)$ *of precision*

Proof. We first remark that for any $1 \leq j \leq d$, the map $i \mapsto \max_j N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)$ is non-increasing, and therefore that $i \mapsto \min_j N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)$ is non-decreasing.

Now, Theorem 5.1.1 implies that the Archimedean embeddings are balanced so that we have for all i:

$$\frac{\max_{\sigma:\mathbf{K}_h\to\mathbf{C}, R_{j,j}^{(i)} \in R^{(i)}} \left|\sigma\left(R_{j,j}^{(i)}\right)\right|}{\min_{\sigma:\mathbf{K}_h\to\mathbf{C}, R_{j,j}^{(i)} \in R^{(i)}} \left|\sigma\left(R_{j,j}^{(i)}\right)\right|} \leq 2^{2p + O\left(\sqrt{n_h \log n_h \log\log n_h}\right)},$$

and so that

$$\frac{\max_j N_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j})}{\min_i N_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j})} = 2^{n_h\left(2p + O\left(\sqrt{n_h \log n_h \log\log n_h}\right)\right)}.$$

Therefore, by combining this bound with the result of Theorem 5.3.3, after the call to **Size-Reduce**, the condition number of $M^{(i)}$ is bounded by

$$2^{2p + O\left(\sqrt{n_h \log n_h \log\log n_h} + d \log n_h\right)}$$

and the computation requires a $O(p)$ bits of precision, with error bounded by 2^{-p}. ∎

5.3.4.2. *Complexity of the* **Lift** *procedure.* With the bounds given by Theorem 5.3.3 we are now able to bound the complexity of the lift procedure described in Algorithm 21.

Lemma 5.3.3 (Quasilinearity of **Lift**). *Let* $\mathbf{K}$ *be the cyclotomic field of conductor* $f > 0$*, of dimension* $n = \varphi(f)$*. Denote by* r *the largest prime factor of* f*. Let* $a, b \in \mathscr{O}_{\mathbf{K}}$ *and suppose that:*

$$\gcd(N_{\mathbf{K}/\mathbf{Q}}(a), N_{\mathbf{K}/\mathbf{Q}}(b)) = 1 \qquad \text{and} \qquad \|a\|+\|b\| \leq 2^p.$$

Then, the time complexity of the algorithm **G-Euclide** *on the inuput* (a, b) *is a*

$$O(r \log(r) n p \log p)$$

for $p \geq \sqrt{n \log n \log\log n}$*. Consequently, it is quasilinear for* $r \leq \log n$*. The output* (u, v) *verify:*

$$au + bv = 1 \qquad \text{and} \qquad \|u\|+\|v\| \leq 2^{p+O\left(\sqrt{n \log n \log\log n}\right)}.$$

Proof. We use a tower of number fields[4] $\mathbf{L}_h^{\uparrow}$, where $\mathbf{L}_i = \mathbf{Q}[x]/\Phi_{f_i}(x)$ and $f_i/f_{i+1} \leq r$. By trivial induction and multiplicativity of the relative norm map, we know that the input of the recursive call at level i, that is, in $\mathbf{L}_i$ is $N_{\mathbf{L}_h/\mathbf{L}_i}(a), N_{\mathbf{L}_h/\mathbf{L}_i}(b)$. As such, with p_i being the number of bits of the coefficients of the input at level i of the recursion, we have $n_i p_i = \mathrm{O}(n_h p)$. Since computing the automorphisms corresponds to permutation of evaluation of a polynomial, each norm can be computed in time $\mathrm{O}(r \log(r) n_i p_i)$ using a product tree [123].

Now, we have by induction that $1 = \det W = \det V$. With R being the R-part of the QR-decomposition of V we have at any level i in the tower $\mathbf{L}_h^{\uparrow}$:

$$\|R_{2,2}\| = \|1/R_{1,1}\| \leq 2^{\mathrm{O}\left(\sqrt{n_i \log n_i \log\log n_i}\right)},$$

so that the size-reduction implies that

$$\begin{aligned}\|M\| &\leq N_{\mathbf{L}_i/\mathbf{Q}}(R_{1,1})^{\frac{1}{n_i}} 2^{\mathrm{O}\left(\sqrt{n_i \log n_i \log\log n_i}\right)} \\ &= \left(n_h\|a\|+n_h\|b\|\right)^{\frac{n_h}{n_i}} 2^{\mathrm{O}\left(\sqrt{n_i \log n_i \log\log n_i}\right)}.\end{aligned}$$

Hence, the output coefficients are also stored using $\mathrm{O}(n_h p/n_i)$ bits. The complexity when $n_0 = 1$, i.e. the **ExGcd** base case, is classically in $\mathrm{O}(p_0 \log p_0)$. Summing along all complexities gives:

$$\mathrm{O}\left(n_h p \log(n_h p) + \sum_{i=1}^{h} r \log(r) n_i p\right) = \mathrm{O}(n_h p \log p + r \log(r) n_h p \log n_h)$$

which simplifies to a $\mathrm{O}(r \log(r) n p \log p)$. ∎

4 Note that this tower is not same as the one used in the whole reduction process. The two towers are indeed constructed independently to optimize the global running time.

5.3.4.3. *Complexity of the top-level.* Now that we have analyzed the complexity and the output quality of each "atomic" parts, we can examine the complexity of the top-level of the algorithm **Reduce**—that is to say its complexity without counting the recursive calls.

Proposition 5.3.2. *Suppose that the following conditions are fulfilled:*

$$\min_{\sigma:\mathbf{K}_h\to\mathbf{C},R_{i,i}^{(1)}\in R^{(1)}}\left|\sigma(R_{i,i}^{(1)})\right|\geq 2^{-p},\quad \alpha=\log^{O(1)}(n_h)$$

$$d\log n_h+\sqrt{n_h\log n_h\log\log n_h}<p.$$

Then, the complexity at the top-level of the algorithm is a $O\left(d^5 n_h p\log p\right)$.

Proof. **Base case: $\mathbf{K}_h=\mathbf{Q}$:** This is a consequence of the analysis of Schönhage's fast reduction [148].

General case: Using Corollary 5.3.1, the number of rounds is $\rho=\mathrm{O}\left(d^2\log p\right)$. By Lemma 5.3.3 the complexity of **Lift**is quasilinear. Thus, the complexity of each round is dominated by the computation of the QR decomposition and the size-reduction. By Theorem 5.3.3, this complexity is a $\mathrm{O}\left(d^3 n_h p/\log p+d^2 n_h p\right)$, yielding a global complexity of $\mathrm{O}\left(d^5 n_h p+d^4 n_h p\log p\right)=\mathrm{O}\left(d^5 n_h p\log p\right)$.

■

5.3.4.4. *Bounding the precision at each level.* We now bound the precision used in the recursive calls at the top-level of the **Reduce**algorithm:

Lemma 5.3.4. *The sum of all bit sizes used in the recursive calls at the top-level is* $O\left(d^2 p\right)$*, when subjected to the conditions:*

$$\min_{\sigma:\mathbf{K}_h\to\mathbf{C},R_{i,i}^{(1)}\in R^{(1)}}\left|\sigma(R_{i,i}^{(1)})\right|\geq 2^{-p}\qquad d\log n_h+\sqrt{n_h\log n_h\log\log n_h}<p.$$

Proof. Recall that the potential of the basis is defined as

$$\Pi=\sum_{j=1}^{d}(d-j)\log(N_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j})),$$

which is in $\mathrm{O}\left(n_h d^2 p\right)$ by assumption on p. Let $1\leq j\leq d$, then the reduction algorithm is about to perform a local reduction of the projected sublattice (r_j,r'_{j+1}), as presented in Paragraph 5.2.1.4, two cases can occur:

- Either $N_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j}^{(i)})\leq\min\left(2^{\alpha n_h^2}\sqrt{N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)},N_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right)$, and as mentioned in Paragraph 5.3.3.3 the local reduction is not performed. We can consider that we use here a zero precision call.
- Either a local reduction is actually performed and by the result of Section 3.3.1, we can use a precision in $\mathrm{O}\left(p_{i,j}\right)$ with:

$$p_i=\log\left(\frac{\max_k\sigma_k(R_{j,j}^{(i)})}{\min_k\sigma_k(R_{j+1,j+1}^{(i)})}\right)$$

to represent the projected lattice. Let now set

$$L = \frac{\log(N_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}/R^{(i)}_{j+1,j+1}))}{n_h}.$$

The precision $p_{i,j}$ is, thanks to the unit rounding Theorem 5.1.1 a

$$\mathrm{O}\left(L + \sqrt{n_h \log n_h \log\log n_h}\right) = \mathrm{O}(L),$$

by hypothesis. The reduction of this truncated matrix yields a unimodular transformation, represented with precision $\mathrm{O}(p_{i,j})$, which when applied to the actual basis matrix implies that Π decreases by a term at least:

$$\delta_{i,j} = n_h \left\lfloor \frac{L}{2} - \alpha n_h \right\rfloor - 2^{-\Omega(p)}$$

by Heuristic 5.3.1 and Theorem 3.3.2. Let us bound the ratio $p_{i,j}/\delta_{i,j}$:

$$\frac{p_i}{\delta_i} = \frac{L + \mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}\right)}{\left(\frac{L}{2} - \alpha n_h\right) n_h - 2^{-\Omega(p_{i,j})}} = \frac{1 + \mathrm{O}\left(\frac{\sqrt{n_h \log n_h \log\log n_h}}{L}\right)}{\frac{n_h}{2} - \frac{\alpha n_h^2}{L} - \frac{2^{-\Omega(p_{i,j})}}{2L}}.$$

Now recall that

$$N_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}) \geq 2^{2(1+\varepsilon)\alpha n_h^2} N_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j+1,j+1})(1 - 2^{-\Omega(p_{i,j})}),$$

the multiplicative error term coming from the precision at which the values of the $R^{(i)}_{j,j}$ and $R^{(i)}_{j+1,j+1}$ are approximated at runtime. Thus, we have:

$$\sqrt{n_h \log n_h \log\log n_h}/L = \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right),$$

and

$$\alpha n_h^2/L \leq \frac{n_h}{2(1+\varepsilon)}.$$

As such we have:

$$\frac{p_{i,j}}{\delta_{i,j}} \leq \frac{1 + \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)}{\frac{n_h \varepsilon}{1+\varepsilon} + \mathrm{o}(1)}.$$

But then, $\delta_{i,j} = \Omega(n_h \varepsilon p_{i,j})$.

The potential is always a sum of non-negative terms, so $\sum_{i,j} \delta_{i,j} \leq \Pi$. The sum of the precision for the calls can thus be bounded by $\mathrm{O}\left(\frac{\varepsilon}{(1+\epsilon)} \frac{\Pi}{n_h}\right) = \mathrm{O}(d^2 p)$, since $\varepsilon = \frac{1}{2}$, which concludes the proof. ∎

Eventually we can prove the general complexity of the algorithm:

Proof of Theorem 5.3.1. The first step of the proof consists in selecting a suitable tower of subfields, for which the relative degrees are chosen to optimize the complexity of the whole reduction. We choose a tower of cyclotomic subfields $\mathbf{K}_h^{\uparrow} = (\mathbf{Q} = \mathbf{K}_0 \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h)$ with $[\mathbf{K}_i : \mathbf{Q}] = n_i$ and $n_{i+1}/n_i = r_i$ which satisfies $r_i/n_{i+1}^{1/5} \in [1; \log f]$, so that $h = \mathrm{O}(\log\log n)$. This always exists as f is log-smooth. We can set $\alpha_i = 4^{h-i+1}$ to satisfy the conditions of Lemma 5.3.4 while making Heuristic 5.3.1 practically possible. By definition of the value set for p we have $p = \mathrm{O}(B)$. And it of course satisfies the requirements of Proposition 5.3.2. Note that by the choices of local precision made in the proof Lemma 5.3.4, a simple induction shows that at each level of the recursion the local precision fulfills the condition of Lemma 5.3.4, by the exact choice of the $p_{i,j}$'s. A by product of this induction asserts that the sum of the precision used in *all* the recursive calls needed to reduce a projected lattice at level i is a

$$\mathrm{O}\left(p\prod_{j=1}^{i-1}\mathrm{O}\left(r_j^2\right)\right) = 2^{\mathrm{O}(i)}B\left(\frac{n}{n_i}\right)^2.$$

Then, since by Proposition 5.3.2 the complexity of the top-level call at level i is a $\mathrm{O}\left(r_i^5 n_i p\log(p)\right) = \mathrm{O}\left(r_i^5 n_i B\log(B)\right)$. Hence the total complexity at level i is $r_i^5/m_i \cdot n^2 B\log(Bn)2^{\mathrm{O}(i)} = n^2 B\log(B)\log^{\mathrm{O}(1)} n$. Summing over all the levels retrieves the announced result. ∎

An important point is that all recursive calls can be computed in parallel, and as most of the complexity is in the leaves, this leads to an important practical speed-up. We conjecture that when the number of processors is at most $n/\log^{\mathrm{O}(1)} n$, the speed-up is linear.

5.4 SYMPLECTIC LATTICES

5.4.1 On symplectic spaces and symplectic groups

In the following, we very briefly introduce the linear theory of symplectic geometry and establish all along this presentation the parallel between the Euclidean and Symplectic geometries.

5.4.1.1. *Definitions.* A *symplectic space* is a finite dimensional vector space E endowed it with an antisymmetric bilinear form $J : E \times E \to E$. We can define a natural orthogonality relation between vectors $x, y \in E$ as being $J(x, y) = 0$. The linear transformations of E letting the symplectic structure J invariant is a group, called the J-symplectic group (or symplectic group if the context makes J clear). This group plays a similar role to the *orthogonal group* for Euclidean spaces.

5.4.1.2. *Darboux bases.* However on the contrary to Euclidean spaces, a symplectic space does not possess an orthogonal basis, but instead a basis

$$e_1, \ldots, e_d, f_1, \ldots, f_d,$$

so that for any indices $i < j$ we have

$$J(e_i, e_j) = 0, J(f_i, f_j) = 0, J(e_i, f_j) = 0$$

and $J(e_i, f_i) > 0$. It implies in particular that any symplectic space has even dimension. We have seen that it is easy to transform any basis of a Euclidean space in an orthogonal basis using the Gram-Schmidt orthogonalization process. This iterative construction is easily adapted to the symplectic case.

5.4.1.3. *Symplectic lattice, size reduction.* We can now easily adapt the definition of a lattice to the symplectic setting:

Definition 5.4.1. *A* symplectic lattice Λ *is a finitely generated free* $\mathbf{Z}$*-module, endowed with a symplectic form* J *on the rational vector space* $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$.

As mentioned in Paragraph 5.2.1.5, an important tool to reduce lattices is the *size-reduction* procedure, which can be viewed as a discretization of the Gram-Schmidt orthogonalization. It aims at reducing the size and the condition number of the lattice basis. When dealing with symplectic symmetries, we can also discretize the process to obtain a basis which is close to a Darboux basis.

As we generalized the lattice formalism to $\mathscr{O}_{\mathbf{L}}$-modules in number fields, we can generalize straightforwardly the notions of symplectic lattices to the algebraic context. Using the work presented in Section 5.2, we aim at providing a fast reduction algorithm for $\mathscr{O}_{\mathbf{L}}$-modules using these symplectic considerations.

5.4.1.4. *Towards an improved algorithmic size-reduction.* The specificities of the symplectic symmetry and of the evoked symplectic size-reduction enable a faster algorithm.

Indeed, we demonstrate that a local reduction within the first half of the matrix can be applied directly to the second half. This almost divides by two the overall complexity *at each descent.*

In the rest of this section, we generalize the work of Gama, Howgrave-Graham and Nguyen [55] on the use of symplectic symmetries lattices within the reduction process. In particular, we show that such techniques can be used for all towers of number fields, and instead of an overall constant factor improvement, we can gain a constant factor at each floor of the tower and then cumulate them. Lattice reduction algorithms hinge on the two following facts:

> **Size reduction:** We can control the bit size without changing the Gram-Schmidt norms.
>
> **Local reduction:** Any two consecutive Gram-Schmidt norms can be made similar.

We therefore have to show that these two parts can be done while preserving the symplectic property.

5.4.2 J-Symplectic group and compatibility with extensions

In all the following we fix an *arbitrary* tower of number fields

$$\mathbf{L}_h^{\uparrow} = (\mathbf{Q} = \mathbf{L}_0 \subset \mathbf{L}_1 \subset \cdots \subset \mathbf{L}_h).$$

For any $1 \leq i \leq h$ we denote by d_h the relative degree of $\mathbf{L}_h$ over $\mathbf{L}_{h-1}$. On any of these number fields, we can define a simple symplectic form, which derives from the determinant form:

Definition 5.4.2. *Let* $\mathbf{L}$ *be a field, and set* J *to be an antisymmetric bilinear form on* $\mathbf{L}^2$. *A matrix* $M \in \mathbf{L}^{2\times 2}$ *is said to be* J-symplectic *(or simply symplectic if there is no ambiguity on* J*) if it lets the form* J *invariant, that is if* $J \circ M = J$.

Let us instantiate this definition in one of the fields of the tower $\mathbf{L}_h^{\uparrow}$ on the 2×2-determinant form. Let J_h be the antisymmetric bilinear form on $\mathbf{L}_h^2$ which is given as the determinant of 2×2 matrices in $\mathbf{L}_h$, i.e.

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0.$$

Remark. *In the presented case,* M *is* J_h*-symplectic iff* $\det M = 1$.

Notice that we can always scale a basis so that this condition is verified.

We descend the form J_h to $\mathbf{L}_{h-1}$ by composition with a non-trivial linear form $\mathbf{L}_h \to \mathbf{L}_{h-1}$, for instance by using the relative trace, that is $J'_h = \mathrm{tr}_{\mathbf{L}_h/\mathbf{L}_{h-1}} \circ J_h$. We then extend the definition of symplectism to $\mathbf{L}_{h-1}^{2d_h}$ by stating that a $2d_h \times 2d_h$ matrix M' is symplectic if it preserves the J'_h form, that is if $J'_h \circ M' = J'_h$. This construction is tailored to be compatible with the descent of a matrix to $\mathbf{L}_{h-1}$ in the following sense:

Lemma 5.4.1. *Let* M *be a* 2×2 *matrix over* $\mathbf{L}_h$ *which is* J_h*-symplectic, then its descent* $M' \in \mathbf{L}_{h-1}^{2d_h \times 2d_h}$ *is* J'_h*-symplectic.*

5.4.3 Towards module transformations compatible with J-symplectism

Before exposing the transformation matrices in our size-reduction process of symplectic lattices, we give an insight on these techniques coming from the Iwasawa decomposition of Lie groups.

5.4.3.1. *On the Iwasawa decomposition.* The *Iwasawa decomposition* is a factorization of any semisimple Lie group in three components, which generalizes the decomposition of $\mathrm{GL}(n, \mathbf{R})$ in the product KAN where $K = O(n, \mathbf{R})$ is the orthogonal group, A is the group of diagonal matrices with positive coefficients and N is the unipotent group consisting of upper triangular matrices with 1s on the diagonal. This decomposition of $\mathrm{GL}(n, \mathbf{R})$ arises directly from the Gram-Schmidt decomposition of any real matrix and

extracting the diagonal of its R part. The $J-$symplectic group defined here is a semisimple Lie group and thus is subject to Iwasawa decomposition. We aim at using an *effective* version of the Iwasawa decomposition. In order to compute effectively such a decomposition, we need to find a generating set of *elementary* transformations over bases, which generalizes the operators of transvections and swaps in the general linear case.

We start by treating a simpler case, suited for cyclotomic extensions: the *Kummer-like extensions*. The general case can be done in a similar way, but requiere to be careful of the ramification of places in the extemsion. We discuss this problem in Appendix 5.

5.4.3.2. *A simple case: Kummer-like extensions* $\mathbf{L}[X]/(X^{d_h}+a)$. We define R_{d_h} as the reverse diagonal of 1 in a square matrix of dimension d_h.

In this section, we use the notation A^s as a shorthand for $R_{d_h}A^TR_{d_h}$, which corresponds to the reflection across the antidiagonal, that is exchanging the coefficients $A_{i,j}$ with A_{d_h+1-i,d_h+1-j}. We proceed here by adapting the work of Sawyer [143]. Suppose that the defining polynomial of $\mathbf{L}_h/\mathbf{L}_{h-1}$ is $X^{d_h}+a$. Recall that J_h is the 2×2-determinant form over $\mathbf{L}_h^2$. We can compose it by the linear form

$$\left|\begin{array}{rcl}\mathbf{L}_h\cong\mathbf{L}_{h-1}[X]/(X^{d_h}+a) & \longrightarrow & \mathbf{L}_{h-1}\\ y & \longmapsto & \mathrm{tr}_{\mathbf{L}_h/\mathbf{L}_{h-1}}(\frac{Xy}{d_h a})\end{array}\right.,$$

to construct the matrix J'_h, which now becomes

$$J'_h=\begin{pmatrix}0 & R_{d_h}\\ -R_{d_h} & 0\end{pmatrix}$$

in the power basis. In this particular setting we retrieve the instantiation of [55]. In particular:

Lemma 5.4.2. *Fix a basis of the symplectic space where the matrix corresponding to* J'_h *is* $\begin{pmatrix}0 & R_{d_h}\\ -R_{d_h} & 0\end{pmatrix}$. *Then, for any* M *a* J'_h*-symplectic matrix and* QR *its QR decomposition, both* Q *and* R *are* J'_h*-symplectic.*

Proof. Direct from the explicit Iwasawa decomposition given by [143]. ∎

Lemma 5.4.3 (Elementary J'_h-symplectic matrices)**.**

- *For any* $A\in GL(d_h,\mathbf{L}_h)$,

$$\begin{pmatrix}A & 0\\ 0 & A^{-s}\end{pmatrix}$$

is J'_h*-symplectic.*

- *For any* $A \in GL(2, \mathbf{L}_h)$ *with* $\det A = 1$ *the block matrix*

$$\begin{pmatrix} \mathrm{Id}_{d_h-1} & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & \mathrm{Id}_{d_h-1} \end{pmatrix}$$

is J'_h *symplectic.*

Proof. By direct computation. ∎

We now turn to the shape of triangular J'_h symplectic matrices.

Lemma 5.4.4. *Block triangular symplectic matrices are exactly the matrices of the form*

$$\begin{pmatrix} A & AU \\ 0 & A^{-s} \end{pmatrix}$$

where $U = U^s$.

Proof. Let $M = \begin{pmatrix} A & U \\ 0 & B \end{pmatrix}$ a block triangular matrix. By Lemma 5.4.3, the action of the block diagonal matrices $\begin{pmatrix} A & 0 \\ 0 & A^{-s} \end{pmatrix}$ by left multiplication preserves the J'_h-symplectic group, so that without loss of generality we can suppose that A is the identity matrix. Identifying the blocks of $M^T J'_h M = J'_h$ yields two relations:

- $R_{d_h} B = R_{d_h}$, entailing $B = \mathrm{Id}_{d_h}$,
- $B^T R_{d_h} U - U^T R_{d_h} B = 0$, so that $R_{d_h} U = U^T R_{d_h}$, and as such $U = U^s$.

∎

5.4.3.3. *Size-reduction of a* J'_h*-symplectic matrix.* A direct consequence of Lemma 5.4.3 is that the local reductions occurring during the reduction, that is swaps and transvections can preserve the J'_h-symplectism by using the corresponding previous constructions.

Consider X a J'_h-symplectic matrix, we want to efficiently *size-reduce* X using the symmetries existing by symplectism. Let first take the R part of the QR-decomposition of X and make appear the factors A and U as in Lemma 5.4.4.

Then we can focus on the left-upper matrix A and size-reducing it into a matrix A'. Each elementary operations performed is also symmetrically performed on A^s to retrieve $(A')^s$. Eventually the size reduction is completed by dealing with the upper-right block, which is done by performing a global multiplication by

$$\begin{pmatrix} \mathrm{Id}_{d_h} & -\lfloor U \rceil \\ 0 & \mathrm{Id}_{d_h} \end{pmatrix}.$$

The corresponding algorithm is given in Algorithm 23, and uses the "classical" **Size-Reduce** procedure as a subroutine. The recursive reduction algorithm using the symplectic structure is then the exact same algorithm as Algorithm 22, where the size-reduction call of line 4 is replaced by **Symplectic-Size-Reduce**.

Algorithm 23 — Symplectic-Size-Reduce

Input : R-factor of the QR decomposition of a J'_h-symplectic matrix $M \in \mathcal{O}^{d\times d}_{\mathbf{L}_h}$

Output : A J'_h-symplectic unimodular transformation U representing the size-reduced basis obtained from M.

1 Set A, U such that $\begin{pmatrix} A & AU \\ 0 & A^{-s} \end{pmatrix} = R$

2 $V \leftarrow$ **Size-Reduce**(A)

3 **return** $\begin{pmatrix} V & -V\lfloor U \rceil \\ 0 & V^{-s} \end{pmatrix}$

The size reduction property on A' implies that both A' and A'^{-1} are small, and therefore it is easy to check that the same is true for the now reduced R' and of course for the corresponding size reduction of the matrix X itself.

This approach admits several algorithmic optimizations:

- Only the first half of the matrix R is actually needed to perform the computation since we can retrieve the other parts. Indeed, with the equation $QR = X$, R is upper triangular and it only depends on the first half of Q.
- Further, we compute only the part above the antidiagonal of AU. This is actually enough to compute the part above the antidiagonal of the matrix $A^{-1}(AU)$, which is persymmetric[5].
- An interesting implication is that since we need to compute only half of the QR decomposition, we need (roughly) only half the precision[6].

5.4.4 Improved complexity

We analyze the algorithm of the previous section with the improvements of Paragraph 5.4.3.3. The notation used in this section are the same as in Section 5.3, with the notable exception that we may use here a large ε—recall that it was fixed to $1/2$ in all of Section 5.3. We also assume that $\alpha >$

5 We call persymmetric a square matrix which is symmetric with respect to the northeast-to-southwest diagonal.

6 Not using the Cholesky decomposition also halves the precision needed.

$\sqrt{\log n_h \log\log n_h}$ for the sake of simplicity. We use here the modified potential where we consider only the first half of the matrix:

$$\Pi = \sum_{i=1}^{d_h} (d_h + 1 - i) \log N_{\mathbf{L}_h/\mathbf{Q}}(R_{i,i}).$$

To complete the proof we need an *experimentally validated heuristic* on the repartition of the potential during the reduction.

Heuristic 5.4.1. *The potential Π is, at the end of* **Reduce**, *always larger than the potential of an orthogonal matrix with the same volume.*

Remark. *This heuristic hinges on the fact the sequence of $N_{\mathbf{L}_h/\mathbf{Q}}(R_{i,i})$ is non-increasing, which is always the case in practice for random lattices.*

We now give a better bound on the increase in bit sizes, which is a refinement of Lemma 5.3.4. The proof is done in the exact same manner.

Lemma 5.4.5. *Suppose the input matrix M is a descent of a 2×2 triangular matrix $\begin{pmatrix} u & v \\ 0 & w \end{pmatrix}$, where the diagonal elements have been balanced in the sense of Theorem 5.1.1. Under Heuristic 5.4.1, the sum of all bit sizes used in the recursive calls at the top-level is at most*

$$p d_h^2 \left(1 + \frac{1}{\varepsilon}\right) \left(\frac{1}{2} + \frac{1}{d_h} + O\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)\right),$$

with

$$p = \log \frac{\max_{\sigma:\mathbf{L}_h \to \mathbf{C}, R_{i,i} \in R} \sigma(R_{i,i})}{\min_{\sigma:\mathbf{L}_h \to \mathbf{C}, R_{i,i} \in R} \sigma(R_{i,i})} \geq n_h d_h,$$

where the σ runs over the possible field embeddings and the $R_{i,i}$ are the diagonal values of the R part of the QR-decomposition of M.

Proof. Without loss of generality, up to scaling, we can assume that

$$N_{\mathbf{L}_{h+1}/\mathbf{Q}}(u) N_{\mathbf{L}_{h+1}/\mathbf{Q}}(w) = N_{\mathbf{L}_h/\mathbf{Q}}\left(\prod_i R_{i,i}\right) = 1.$$

Therefore, with our choice of p, we have at the beginning

$$\|R_{i,i}\| \leq \|u\| \in 2^{p/2 + O\left(\sqrt{n_h d_h \log(n_h d_h) \log\log(n_h d_h)}\right)}.$$

Thus we have :

$$\begin{aligned} \Pi &= \frac{n_h d_h (d_h + 1)}{4} \left(p + O\left(\sqrt{n_h d_h \log(n_h d_h) \log\log(n_h d_h)}\right)\right) \\ &= \frac{n_h d_h (d_h + 1)}{4} p \left(1 + O\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)\right), \end{aligned}$$

since by hypothesis, $p > n_h d_h$. And then by, Heuristic 5.4.1, we have $\Pi \geq 0$ at the end of the calls. When performing local reductions, as in the proof of Lemma 5.3.4, two cases can occur:

- Either $N_{\mathbf{L}_h/\mathbf{Q}}(R^{(i)}_{j,j}) \leq 2^{2(1+\varepsilon)\alpha n_h^2} N_{\mathbf{L}_h/\mathbf{Q}}(R^{(i)}_{j+1,j+1})$, and as mentioned in Paragraph 5.3.3.3 the local reduction is not performed, so that we can consider that we use here a zero precision call.
- Either a local reduction is actually performed and by the result of Section 3.3.1, we can use a precision in $\mathrm{O}(p_{i,j})$ with:

$$p_{i,j} = \log\left(\frac{\max_k \sigma_k(R^{(i)}_{j,j})}{\min_k \sigma_k(R^{(i)}_{j+1,j+1})}\right),$$

Let now set

$$L = \frac{\log(N_{\mathbf{L}_h/\mathbf{Q}}(R^{(i)}_{j,j}/R^{(i)}_{j+1,j+1}))}{n_h}.$$

The value $p_{i,j}$ is, thanks to the unit rounding Theorem 5.1.1 a

$$L + \mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}\right),$$

by hypothesis. The reduction of this truncated matrix yields a unimodular transformation, represented with precision $\mathrm{O}(p_{i,j})$, which when applied to the actual basis matrix implies that Π decreases by a term at least:

$$\delta_{i,j} = n_h\left\lceil \frac{L}{2} - \alpha n_h \right\rfloor - 2^{-\Omega(p)}$$

by Heuristic 5.3.1 and Theorem 3.3.2. Let us bound the ratio $p_{i,j}/\delta_{i,j}$:

$$\frac{p_i}{\delta_i} = \frac{L + \mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}\right)}{\left(\frac{L}{2} - \alpha n_h\right) n_h - 2^{-\Omega(p)}} = \frac{1 + \frac{\mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}\right)}{L}}{\frac{n_h}{2} - \frac{\alpha n_h^2}{L} - \frac{2^{-\Omega(p)}}{2L}}.$$

Now recall that $N_{\mathbf{L}_h/\mathbf{Q}}(R^{(i)}_{j,j}) \geq 2^{2(1+\varepsilon)\alpha n_h^2} N_{\mathbf{L}_h/\mathbf{Q}}(R^{(i)}_{j+1,j+1})(1 - 2^{-\Omega(p)})$, the multiplicative error term coming from the precision at which the values of the $R^{(i)}_{j,j}$ and $R^{(i)}_{j+1,j+1}$ are approximated at runtime. Thus we have:

$$\sqrt{n_h \log n_h \log\log n_h}/L = \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right),$$

and

$$\alpha n_h^2/L \leq \frac{n_h}{2(1+\varepsilon)}.$$

As such we have:

$$\frac{p_{i,j}}{\delta_{i,j}} \leq \frac{1 + \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)}{\frac{n_h \varepsilon}{1+\varepsilon} + \mathrm{O}(1/n_h)}.$$

The sum of precisions is therefore multiplied by

$$d_h^2\left(1 + \frac{1}{\varepsilon}\right)\left(\frac{1}{2} + \frac{1}{2d_h} + \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)\right),$$

which finishes the proof. ∎

We can now collect all the calls at each level to compute the global complexity, for refining Theorem 5.3.1:

Theorem 5.4.1. *Select an integer f a power of $q = O(\log f)$ and let $n = \varphi(f)$. The complexity for reducing matrices M of dimension two over $\mathbf{K} = \mathbf{Q}[x]/\Phi_f(x)$ with B the number of bits in the input coefficients is heuristically*

$$\tilde{O}\left(n^{2+\frac{\log((1/2+1/2q)(1+1/\varepsilon))}{\log q}} B\right)$$

and the first column of the reduced matrix has coefficients bounded by

$$\exp\left(O\left(n^{1+\frac{\log((1+\varepsilon)\frac{2q-1}{q})}{\log q}}\right)\right)\left|N_{\mathbf{L}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{2n}}.$$

Proof. The proof is now exactly the same as for Theorem 5.3.1. We select a tower of cyclotomic subfields $\mathbf{L}_h^{\uparrow}$ with $\mathbf{L}_0 = \mathbf{Q}$, $[\mathbf{L}_i : \mathbf{Q}] = n_i$, $n_{i+1}/n_i = d_i = q$ for $i < h$ and $\mathbf{L}_h = \mathbf{K}$ with $h = \log f / \log q$. We remark that we can take

$$\alpha_i = n_i\left((1+\varepsilon)\frac{2q-1}{q}\right)^i$$

and all our previous assumptions are fulfilled.

The complexity at the level i is $\mathrm{O}\left(q^5 n_i p \log(Bn)\right)$ for precision p but the sum on the precision over all calls is a:

$$\mathrm{O}\left(B\prod_{j>i}\left(1+\frac{1}{\varepsilon}\right)\left(\frac{1}{2}+\frac{1}{2q}+\mathrm{O}\left(\sqrt{\frac{\log n_i \log\log n_i}{n_i}}\right)\right)d_j^2\right),$$

which simplifies in

$$\mathrm{O}\left(B\left(\frac{n}{n_i}\right)^2\left(\frac{(1+\frac{1}{\varepsilon})(q+1)}{2q}\right)^{h-i}\right).$$

Summing over all i gives the result. ∎

Selecting $\varepsilon = \log n$, and combining with the analysis of previous section yields:

Corollary 5.4.1. *Select an integer f a power of $q = O(\log f)$ and let $n = \varphi(f)$. The complexity for reducing matrices M of dimension two over $\mathbf{K} = \mathbf{Q}[x]/\Phi_f(x)$ with B the number of bits in the input coefficients is heuristically*

$$\tilde{O}\left(n^{2+\frac{\log(1/2+1/2q)}{\log q}} B\right) + n^{O(\log\log n)}$$

and the first column of the reduced matrix has coefficients bounded by

$$2^{\tilde{O}(n)}\left|N_{\mathbf{L}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{2n}}.$$

Clearly, for $B = n^{\omega(1)}$, we can choose $\varepsilon = \omega(1)$ and get a running time of

$$n^{2+\frac{\log(1/2+1/2q)}{\log q}+\mathrm{o}(1)} B.$$

5.5 OPTIMIZATIONS AND PRACTICAL CONSIDERATIONS

The framework introduced in the previous sections have been implemented and tested. This section details various optimizations, implementation choices, improvement directions, as well as gives an experimental assessment on the heuristics used in the complexity proofs.

5.5.1 On the choice of the base case

Let $h > 0$ be a non-negative integer. The setting of the reduction is a tower of power-of-two cyclotomic fields $\mathbf{L}_h^{\uparrow} = (\mathbf{Q} = \mathbf{L}_0 \subset \mathbf{L}_1 \subset \cdots \subset \mathbf{L}_h)$.

5.5.1.1. *Stopping the reduction before hitting* $\mathbf{Z}$. As stated in Theorem 5.3.1, the approximation factor increases quickly with the height of the tower. However, if we know how to perform a reduction over a number field above $\mathbf{Q}$, say $\mathbf{L}_1$ for instance, directly, then there is no need to reduce up to getting a $\mathbf{Z}$-module and we instead stop at this level. Actually, the largest the ring, the better the approximation factor becomes and the more efficient is the whole routine. As seen in Chapter 3, it is possible to come up with a *direct* reduction algorithm for an algebraic lattice when the underlying ring of integer is *norm-Euclidean*. Hence a natural choice would be $\mathbf{Z}[x]/(x^n + 1)$ with $n \leq 8$ as these rings are proved to be norm-Euclidean.

5.5.1.2. *The ring* $\mathbf{Z}[x]/(x^{16} + 1)$. However, it turns out that while $\mathbf{L} = \mathbf{Z}[x]/(x^{16} + 1)$ is not norm-Euclidean, we can still use this as our base case. As such, we need to slightly change the algorithm in case of failure of the regular LLL algorithm. Given a, b, we use the *randomized* unit rounding of $\sqrt{\{\mu\}}$ computed by Theorem 5.1.1 with $\mu = a/b$, which gives a unit u such that $u^2\{\mu\}$ is round. We accept the change if

$$N_{\mathbf{L}/\mathbf{Q}}(a - b(\lfloor \mu \rfloor + \lfloor u\{\mu\} \rceil u^{-1})) < N_{\mathbf{L}/\mathbf{Q}}(a)$$

and restart up to a hundred times if it fails.

This algorithm restarts on average 0.7 times and fails every 50000 times. On failure, one can for example use a more complicated approach; but as long as the number of bits is not gigantic, we can simply stop there since the other reductions around the two Gram-Schmidt norms will randomize everything and the algorithm can smoothly continue. The terms a, b tend to slowly accumulate a unit contribution when $n \geq 4$, and it is therefore needed to rebalance them using randomized rounding. For $n = 16$, this happens on average every 50 times.

5.5.1.3. *Comparison between the base fields.* We give in the Table 5.2 the properties of the various possible base cases between the dimension 1 over $\mathbf{Q}$—that is $\mathbf{Q}$ itself—and 16, as described above.

Table 5.2: Lattice reduction with root factor α in dimension d over $\mathbf{Z}$ gives an element of Λ of norm around $\alpha^{d/2}\operatorname{covol}(\Lambda)^{1/d}$. After k steps in the Euclidean algorithm with norm factor β, the norm of the elements is roughly divided by β^k. Both are for random inputs.

Dimension	Root factor	Norm factor
1	1.031	4.6
2	1.036	7.1
4	1.037	17
8	1.049	26
16	1.11	24

Remark (A heuristic optimization of bitsize of the lattice)**.** *In practice, we of course want the base case to be (relatively) fast. A heuristic method, used for instance in the implementation* fpLLL*, consists in applying a divide-and-conquer strategy on the most significant bits: we first reduce the input matrix with half the precision, apply the transition matrix, and reduce the rest with about half the precision. After that, we can run the full reduction one more time to verify that the lattice is indeed reduced.*

5.5.2 Decreasing the approximation factor

In several applications, it is interesting to decrease the approximation factor. Our technique is, at the lowest level of recursion, and when the number of bits is low, to use a LLL-type algorithm. Each time the reduction is finished, we descend the lattice to a lower level where the approximation factor is lower and restart the reduction on this descent.

5.5.3 Lifting a reduction

One might expect that, as soon as the ideal generated by all the $N_{\mathbf{K}/\mathbf{L}}(a_i)$ and $N_{\mathbf{K}/\mathbf{L}}(b_i)$ is $\mathscr{O}_{\mathbf{L}}$, that for most of the small $x \in \mathscr{O}_{\mathbf{K}}$, we would have

$$N_{\mathbf{K}/\mathbf{L}}(\langle a, x\rangle)\mathscr{O}_{\mathbf{L}} + N_{\mathbf{K}/\mathbf{L}}(\langle b, x\rangle)\mathscr{O}_{\mathbf{L}} = \mathscr{O}_{\mathbf{L}}.$$

There is, however, a profusion of counterexamples to this and the algorithm often stumbles on them. This implies that the lift of a short vector can actually be quite large, depending on the norm of the ideal generated by the elements $N_{\mathbf{K}/\mathbf{L}}(\langle a, x\rangle)$ and $N_{\mathbf{K}/\mathbf{L}}(\langle b, x\rangle)$. A solution which practically works is to increase the number of short vectors we consider in the lifting phase: instead of lifting one vector, we lift multiple of them. As such, the lift step never causes problem when we are reducing a random lattice. In our exper-

iments with random lattices, the average number of lifted vectors is around 1.5.

When the lattice is not random, for example with a short planted element, it sometimes completely fails: at each round in the algorithm, the lift will return a long vector even if the recursive reduction found plenty of short ones. While this may not be a problem for some applications – finding a short vector in a NTRU lattice implies an ability to decrypt – it is an important one for others. Our proposed solution to this difficulty is to use a pseudo-basis instead of a basis. Indeed, it is a standard fact that the first element can be lifted into a unimodular pseudo-basis [35, Corollary 1.3.5]. Of course, we need to have a fast ideal arithmetic and to keep the ideals of small norm, which is neither easy nor fast and will be the subject of a future work.

5.5.4 Towards a fully recursive structure

A bottleneck with Algorithm 22 is that each round needs a matrix multiplication, and there are at least d^2 rounds. However, one can notice that each round only make local modifications. As a result, we propose to use a small number D of blocks, typically 4 or 8 suffices, and a round will (recursively) reduce consecutive pairs of dimension d/D. The resulting number of rounds is again $\mathrm{O}(D^2 \log B)$, which gives a top-level complexity of $\mathrm{O}(D^2)$ (equivalent) multiplications. The corresponding algorithm is given in the Algorithm 24. We describe this algorithm with respect to an oracle **Oracle** which computes the base case. One can either use **Schonhage**, the algorithms in the previous or current section, or a recursive call. Thus, this general strategy can be used to reduce Euclidean lattices, as well as large rank algebraic lattices. A precise analysis on its cost would be of interest and is a clear direction of future work.

Algorithm 24 — Recursive Reduce

Input : Basis $M \in \mathcal{O}_{\mathbf{K}}^{d \times d}$ of the $\mathcal{O}_{\mathbf{K}}-$module $\mathcal{M}$
Output : A unimodular transformation $U \in \mathcal{O}_{\mathbf{K}}^{d \times d}$ representing a reduced basis of $\mathcal{M}$.

if $d = 2$ **then return** **Oracle**(M)
for $i = 1$ **to** ρ **do**
  $R \leftarrow$ **Orthogonalize**(M)
  $U_i \leftarrow$ **Seysen-Size-Reduce**(R)
  $(M, R) \leftarrow (M, R) \cdot U_i$
  for $j = 1 + (i \mod 2)$ **to** d **by step of** $2d/D$ **do**
    $U' \leftarrow$ **Reduce**$(R[j : j + 2d/D - 1, j : 2d/D - 1])$
    $(U_i, M) \leftarrow (U_i, M) \cdot \mathrm{Diag}\left(\mathrm{Id}_j, U', \mathrm{Id}_{2d-j-2}\right)$
  end for
end for
return $\prod_{i=1}^{\rho} U_i$ // The product is computed from the end

5.5.5 Other details

The prototype of the program was written in the interpreted language Pari/GP [10]. It uses the native functions for multiplying field elements, which is not at all optimal, and even more so when we multiply matrices. Only the recursive calls were parallelized, and not the Gram-Schmidt orthogonalization nor the size reduction, which limits the speed-up we can achieve in this way. We used the Householder method for the QR-decomposition. The symplectic optimization was used at each step, and was not found to change the quality of the reduction[7].

5.6 APPLICATIONS TO THE GENTRY-SZYDLO ALGORITHM

The fast reduction procedure for cyclotomic ideals can be used to build a fast implementation of the Gentry-Szydlo algorithm [64]. This algorithm retrieves, in polynomial time, a generator of a principal ideal $f\mathcal{O}_{\mathbf{L}}$ given its relative norm $f\overline{f}$ in cyclotomic fields, or more generally in CM fields. This algorithm is a combination of algebraic manipulations of ideals in the field and lattice reduction.

5.6.1 Gentry-Szydlo.

In this section, we briefly recall the crux of the Gentry-Szydlo algorithm [64]. This algorithm aims at solving the following problem, presented in its whole generality:

Problem (Principal ideal problem with known relative norm)**.** *Let* $\mathbf{L}$ *be a CM-field, of conjugation* $x \mapsto \overline{x}$*, and denote by* $\mathbf{L}^+$ *its maximal totally real subfield. Let* $f \in \mathcal{O}_{\mathbf{L}}$ *and set* $\mathfrak{f} = f\mathcal{O}_{\mathbf{L}}$*, the ideal spanned by this algebraic integer.*

> ***Input:*** *The relative norm* $N_{\mathbf{L}^+/\mathbf{Q}}(f) = f\overline{f}$ *and a* $\mathbf{Z}$*-basis of the ideal* $\mathfrak{f}$*.*
>
> ***Output:*** *The element* f*.*

We can use the reduction of an ideal as follows: from $\mathfrak{f}$ and $f\overline{f}$ we start by reducing the $\mathcal{O}_{\mathbf{L}}$-lattice

$$\frac{f\mathcal{O}_{\mathbf{L}}}{\sqrt{f\overline{f}}},$$

and find an element of the shape fx where $x \in \mathcal{O}_{\mathbf{L}}$ and is small. Now we have that:

$$\mathfrak{f} = \frac{f\overline{f}}{\overline{fx}}\overline{x}\mathcal{O}_{\mathbf{L}}$$

7 Gama, Howgrave-Graham and Nguyen [55] found instead that it gave a "smoother (better)" basis, showing a significant difference in their Figure 1. An other version of the paper does not include this comment, and their (perplexing) Figure 1 shows no difference in the exponential decrease of the Gram-Schmidt norms.

We also have $x\overline{x} = \frac{fx\overline{fx}}{f\overline{f}}$ so that we have reduced the problem to the smaller instance $(\overline{x}\mathcal{O}_{\mathbf{L}}, x\overline{x})$.

For the sake of simplicity, we give here the outline of the remaining part of the algorithm for a cyclotomic field of conductor a power of two. The algorithm selects an integer e such that $f^e \mod r$ is known with a large r. Binary exponentiation with the above reduction computes a $x\mathcal{O}_{\mathbf{L}}$ with a short $x \in \mathcal{O}_{\mathbf{L}}$ and such that

$$f^e = Px$$

with P known (and invertible) modulo r and q^k. Now we can deduce $x \mod r$ and since x is small, we know x.

The last step is to extract an e-th root modulo q^k. We choose q such that $q\mathcal{O}_{\mathbf{L}} = \mathfrak{q}\overline{\mathfrak{q}}$ which always exists in power of two cyclotomic fields since $(\mathbf{Z}/2n\mathbf{Z})^\times/\{-1,1\}$ is cyclic. Extracting e-th root modulo $\mathfrak{q}$ is easy, as e is smooth. There are $\gcd(e, q^{n/2}-1)$ such roots, and we can choose q such that for each $p|e$ with p not a Fermat prime, $q^{n/2} \neq 1 \mod p$. If we choose $f \mod \mathfrak{q}$ as a root, then we know $\overline{f} \mod \overline{\mathfrak{q}}$, and we also know $f\overline{f}$ so we can deduce $f \mod \overline{\mathfrak{q}}$. As a result, we know $f \mod q$ and Hensel lifting leads to $f \mod q^k$. For k sufficiently large, we recover f.

We choose e to be the smallest multiple of $2n$, such that r, the product of primes p such that $2n|p-1|e$, is sufficiently large. One can show [96] that $\log e = O(\log n \log\log n)$ is enough and heuristically taking e as the product of n and a primorial reaches this bound.

5.6.2 Faster multiplication using lattice reduction.

The bottleneck of the Gentry-Szydlo algorithm is to accelerate the ideal arithmetic. We represent ideals with a small family of elements over the order of a subfield $\mathcal{O}_{\mathbf{L}}$. One can represent the product of two ideals using the family of all products of generators. However, this leads to a blow-up in the size of the family. A reasonable approach is simply to sample a bit more than $[\mathbf{L} : \mathbf{K}]$ random elements in the product so that with overwhelming probability the ideal generated by these elements is the product ideal itself. It then suffices to reduce the corresponding module to go back to a representation with smaller generators.

An important piece is then the reduction of an ideal itself. Our practical approach is here to reduce a square matrix of dimension $[\mathbf{L} : \mathbf{K}]$, and every two rounds to add a new random element with a small Gram-Schmidt norm in the ideal at the last position.

In our experiment, we reduce up to 1.05^n (respectively 1.1^n) the first ideal to accelerate the powering with $n \leq 512$ (respectively $n = 1024$). The smallest e such that this approximation works at the end was chosen. The other reductions are done with an approximation factor of $2^{n/5}$ (respectively $2^{n/3}$).

We emphasize that the implementation hardly used all cores: for example, the total running time over all cores in the last case was 354 hours.

Table 5.3: Implementation results

Dimension	e	Running time	Processor
256	15360	30 minutes	Intel i7-8650 (4 cores)
512	79872	4 hours	Intel i7-8650 (4 cores)
1024	3194880	103 hours	Intel E5-2650 (16 cores)

The runtime of the first implementation published [17] in dimension 256 was 20 hours. Assuming it is proportional to n^6 leads to an estimate of 10 years for $n = 1024$, or 800 times slower than our algorithm. Our practical results are compiled in Table 5.3.

This algorithm will play a role in the following chapter to speedup a whole algorithm for solving the so-called *principal ideal problem*, as well as in the cryptanalysis of a signature scheme in Chapter 8.

6

THE PRINCIPAL IDEAL PROBLEM

In the previous chapter we devised algorithms for the reduction of algebraic lattices and in particular of so-called ideal lattices. We now make use of these reduction to solve a well-known problem in effective number theory.

Recall that in an arbitrary number field, even though every ideal is generated by two elements, not all of them can be generated by only one.

Example. *Let* $\mathbf{L} = \mathbf{Q}[i\sqrt{5}]$. *The fields embeddings of* $\mathbf{L}$ *into* $\mathbf{C}$ *are on the one hand the identity map and on the other hand the linear map sending* $i\sqrt{5}$ *on* $-i\sqrt{5}$. *Hence the algebraic norm of any element* $\alpha = (a + bi\sqrt{5})$ *is* $N_{\mathbf{LQ}}(\alpha) = a^2 + 5b^2$.

Let now $\mathfrak{a}$ *be the ideal of* $\mathscr{O}_{\mathbf{L}}$ *spanned by the elements* 2 *and* $1 + i\sqrt{5}$. *Let us first show that this ideal is of norm* 2. *We fix the* $\mathbf{Z}$*-basis* $\mathcal{B} = (1, 1 + i\sqrt{5})$ *of* $\mathscr{O}_{\mathbf{L}} = \mathbf{Z}[i\sqrt{5}]$. *Hence, the ideal* $\mathfrak{a}$ *is generated as* $\mathbf{Z}$*-module by* $(2, 2 + 2i\sqrt{5}, 1 + i\sqrt{5}, 2i\sqrt{5} - 4)$, *that is by* $(2, 1 + i\sqrt{5})$. *Thus the corresponding matrix in* $\mathcal{B}$ *is:*

$$M = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix},$$

implying that

$$N(\mathfrak{a}) = \left|\mathscr{O}_{\mathbf{L}}/\mathfrak{a}\right| = \left|\mathbf{Z}^2/M\mathbf{Z}\right| = \det(M) = 2.$$

This ideal is not principal. Indeed, suppose it is generated by an element α. *Then we have* $N(\mathfrak{a})|N(2\mathscr{O}_{\mathbf{L}})$. *Since* $N(2\mathscr{O}_{\mathbf{L}}) = N_{\mathbf{L}/\mathbf{Q}}(2) = 4$, *we deduce that* $N_{\mathbf{L}/\mathbf{Q}}(\alpha)|4$. *Similarly,* $N(\mathfrak{a})|N((1 + i\sqrt{5})\mathscr{O}_{\mathbf{L}})$, *implying that* $N_{\mathbf{L}/\mathbf{Q}}(\alpha)|6$. *As such we have* $N_{\mathbf{L}/\mathbf{Q}}(\alpha)|2$. *As there are no integers of norm equal*[1] *to 2, this is the desired contradiction.*

The *Principal Ideal Problem* (PIP) consists in finding a generator of an ideal in a number field, assuming it is principal.

Example. *Let us fix a* $\mathbf{Z}$*-basis of* $\mathscr{O}_{\mathbf{L}}$, *for instance* $\mathcal{B} = (1, i\sqrt{5})$. *An instance of the principal ideal problem in the field* $\mathbf{L}$ *can be for example: given the ideal* T *represented by the free* $\mathbf{Z}$*-module spanned by* $\begin{pmatrix} 18 & 14 \\ 0 & 2 \end{pmatrix}$ *in* $\mathcal{B}$, *retrieve its generator, up to a unit. In this particular case, an admissible answer is* $\tau = 4 - 2i\sqrt{5}$, *as we can check that* $N_{\mathbf{L}/\mathbf{Q}}(\tau) = 36 = N(T)$.

1 Indeed, for a generic element $\alpha = x + yi\sqrt{5}$ we have $N_{\mathbf{L}/\mathbf{Q}}(\alpha) = x^2 + 5y^2$, which clearly can not be equal to 2.

In degree one—that is over the field $\mathbf{Q}$—the problem is fairly easy, as it reduces to find a generator of an ideal of $\mathbf{Z}$ given by some generators, that is to compute the greatest common divisor of these elements, as $\mathbf{Z}$ is principal. In higher degree, the problem is considered as hard in computational number theory. It appears that its resolution is actually related to another hard problem in this area: the computation of the class groups of the number field, as we need to sample independent relations in the class group. Interestingly, even the (apparently) simpler problem of testing the principality of an arbitrary ideal does not seem easier, as precised by Cohen in [34, Chapter 4] and Thiel in [160, Section 7] (essentially, solving this problem consists in trying to compute the class of the ideal in the class group to compare it with the neutral elements).

More practically, in the field of cryptography, this problem appears in the context of Fully Homomorphic Encryption (FHE) [62], a scheme of encryption where computations are possible over encrypted data. A usual setting for this type of cryptographic schemes are ring of integers of cyclotomic fields. For instance, the security of the scheme presented by Smart and Vercauteren [153] relies on the difficulty of PIP, which consists in finding a short generator of a principal ideal. We discuss the practical implication of the resolution of PIP in the final part of this manuscript, where we demonstrate a total break of the scheme Chapter 8.

From class group computations to PIP

Solving the principal ideal problem essentially requires the computation of the ideal class group, that is the class group of the ring of integers, of the number field $\mathbf{L}$ where the ideals are defined. This approach is described in [34, Algorithm 6.5.10]. The first subexponential algorithm for computing the class group was due to Hafner and McCurley [71]. It applies in the context of imaginary quadratic fields and has been generalized by Buchmann [25] to classes of number fields of fixed degree. In [14], Biasse and Fieker presented an algorithm for computing the class group in subexponential time in arbitrary classes of number fields. This yields a subexponential time algorithm for solving the PIP in arbitrary classes of number fields. In a prime-power cyclotomic field of degree n, the Biasse-Fieker algorithm solves the PIP in time $L_\Delta\left[\frac{2}{3}+\varepsilon\right] \approx 2^{n^{2/3+o(1)}}$, for $\varepsilon > 0$ arbitrarily small.

A quantum polynomial-time algorithm for solving the PIP was also described by Biasse and Song in [16]. In this chapter, we present a subexponential algorithm which solves the PIP in general cyclotomic fields. Its running time in such a field $\mathbf{L}$ is

$$L_\Delta\left[\frac{1}{2}, \frac{\omega}{2\sqrt{2(\omega-1)}}\right] \approx 2^{n^{1/2+o(1)}},$$

with ω being the exponent of the arithmetic complexity of matrix multiplication.

6.1 ADDITIONAL BACKGROUND AND SPECIFIC NOTATIONS

Before diving into the subexponential solution to the principal ideal problem, we reintroduce here complementary notions on smoothness in ideal arithmetic, on the possible embeddings of an ideal and on subexponential complexity.

6.1.1 The L notation of subexponential complexities

When dealing with subexponential complexities, the L notation has become pretty ubiquitous, especially when studying sieve-based algorithms such as factorization or discrete-logarithm algorithms. It has been introduced by Pomerance in [136] to simplify the analysis of some factoring algorithms and has then been refined by Lenstra and Lenstra in [108] who introduced the second constant term.

Given two constants a and c with $\alpha \in [0,1]$ and $c \geq 0$, define the class:

$$\mathrm{L}_n[\alpha, c] = \exp\Big((c + \mathrm{o}(1)) (\log n)^{\alpha} (\log\log n)^{1-\alpha} \Big),$$

where $\mathrm{o}(1)$ is the class of functions vanishing as n tends to infinity. We also encounter the simplified notation $L_n(\alpha)$ when specifying c is superfluous. This class encompasses the functions of polynomial-growth when taking $\alpha = 0$ and the exponential-growth when $\alpha = 1$. The term in

$$\exp\Big(c (\log n)^{\alpha} (\log\log n)^{1-\alpha} \Big)$$

expresses the dominant term, while the

$$\exp\Big(\mathrm{o}(1) (\log n)^{\alpha} (\log\log n)^{1-\alpha} \Big)$$

term hides all of the negligible such as polynomial factors.

Remark. *Let $\mathbf{L} = \mathbf{Q}[\zeta_f]$ the cyclotomic field of conductor f. Then by Theorem 1.6.2, its discriminant Δ satisfies:*

$$\Delta = (-1)^{\varphi(n)/2} \frac{n^{\varphi(n)}}{\prod_{p|n} p^{\varphi(n)/(p-1)}},$$

so that for any $\alpha > 0$ we have

$$\log(L_{\Delta}[\alpha]) = f^{\alpha + o(1)},$$

using the trivial estimate $\varphi(f) < f$. Denoting by Δ^+ the discriminant of its maximal real subfield we have

$$\log(L_{\Delta^+}[\alpha]) = f^{\alpha + o(1)},$$

as $\Delta \geq (\Delta^+)^2$.

6.1.2 Smooth ideals

6.1.2.1. *Smoothness and ideals.* Let us fix a number field $\mathbf{L}$ of discriminant Δ. Recall that any ideal of $\mathscr{O}_{\mathbf{L}}$ can be decomposed as a product of prime ideals and that this decomposition is unique up-to-the ordering of the product.

Definition 6.1.1. *Let $\mathfrak{a}$ be a principal ideal of $\mathscr{O}_{\mathbf{L}}$ and $y > 0$ a non-negative real. The ideal $\mathfrak{a}$ is said to be y-smooth if it is the power-product of prime ideals of norm all bounded by y.*

For our purposes, we need to rely on heuristics to evaluate the smoothness probabilities of ideals. They are generalizations of similar results for integers of Canfield, Erdős and Pomerance [32].

Heuristic 6.1.1. *The probability*[2] *$\mathcal{P}(x, y)$ that an ideal of norm bounded by x is y-smooth satisfies*

$$\mathcal{P}(x, y) \geq e^{-u(\log u)(1+o(1))} \quad \textit{for} \quad u = \frac{\log x}{\log y}.$$

Corollary 6.1.1. *Assuming that $x = L_\Delta[\alpha_1, c_1]$ and $y = L_\Delta[\alpha_2, c_2]$, with $\alpha_1 > \alpha_2$, Heuristic 6.1.1 can be expressed as*

$$\mathcal{P}(x, y) = L_\Delta\left[\alpha_1 - \alpha_2, (\alpha_1 - \alpha_2)\frac{c_1}{c_2}\right]^{-1}.$$

A similar assertion for smoothness of ideals was proved by Seysen [150] in 1985 for the quadratic case, but for arbitrary degree, it remains conjectural, even under the Generalized Riemann Hypothesis. This is one of the reasons why the complexity of the number field sieve (NFS) [110] is still a heuristic estimation.

6.1.2.2. *On smoothness testing.* A natural computational question raised by the definition of smoothness is how to test if an arbitrary ideal $\mathfrak{a}$ is B-smooth for a bound $B > 0$. Following the definition gives the naive approach of factoring $\mathfrak{a}$ in prime ideals and compute the norm of each of them. However, this approach requires to factor the norm of $\mathfrak{a}$ over the integers and as such its complexity depends on $\mathfrak{a}$ and not on B, which can be way smaller than $N(\mathfrak{a})$. Now remark that if $\mathfrak{a}$ is B-smooth, then, in particular, its norm is also B-smooth. Hence, we want to test the smoothness of $N(\mathfrak{a})$. If this norm is not B-smooth we thus know that the ideal is not smooth. In the other case, if we know the prime factors appearing in the norm, it suffices to find the valuations at the prime ideals above them. A way to derive these valuations is explained in [34, Section 4.8.3]. The algorithm described also has a complexity that is polynomial in the extension degree and the size of the prime and henceforth in $\log B$.

2 The probability distribution taken here is the uniform distribution over the *finite* set of ideals of norm bounded by x.

The whole test then reduces to efficiently test an integer for smoothness and to extract its factors if so. This can be done efficiently using a Monte-Carlo approach based on the Elliptic Curve Method (ECM) for integer factorization.

This algorithm has been introduced in 1985 by Lenstra [107]. It is the asymptotically fastest method that has been published for finding relatively small factors of large composites. Given an odd composite integer n to be factored, this method consists in performing arithmetic operations on elliptic curves considered over a finite field $\mathbf{F}_p$ for an unknown prime p dividing n. It finds p if the cardinality of at least one of these curves over the field $\mathbf{F}_p$ is smooth. For this reason, we use curves that are known to have favorable smoothness properties, such as a large torsion group over $\mathbf{Q}$ or a cardinality that is divisible by a fixed factor.

Let N be an integer to test for B-smoothness. The crux of the smoothness testing is to launch ECM on as many curves as we can in time $\mathrm{Poly}(\log N) \cdot \mathrm{L}_B\left[\frac{1}{2}, \sqrt{2}\right]$, which will return the factorization of N if it is B-smooth and "non smooth" otherwise.

Heuristic 6.1.2. *Let $\mathbf{L}$ be a number field, $\mathfrak{a}$ an ideal of $\mathscr{O}_\mathbf{L}$ and $B > 0$. Then, we can test in expected time:*

$$Poly([\mathbf{L} : \mathbf{Q}], \log N(\mathfrak{a})) \cdot L_B\left[\frac{1}{2}, \sqrt{2}\right], \tag{6.1}$$

the B-smoothness $\mathfrak{a}$.

6.1.3 On the possible lattices structure of ideals

Let $\mathbf{L}$ be a number field of degree n and $\mathfrak{a}$ an ideal of $\mathscr{O}_\mathbf{L}$. In Paragraph 3.3.1.3, we saw that since $\mathfrak{a}$ has a natural structure of $\mathscr{O}_\mathbf{L}$-module, it can be endowed with the canonical norm of $\mathbf{L}$ so that it is actually an $\mathscr{O}_\mathbf{L}$-lattice. Thus its direct image over $\mathbf{Z}$ is a $\mathbf{Z}$-lattice, as described in Chapter 3. However in certain cases it can be useful to consider a somehow more "direct" lattice structure by taking the norm of the coefficients appearing in the decomposition in a fixed basis of $\mathscr{O}_\mathbf{L}$. Since this concept will only be developed in the case of the cyclotomic extensions we stick to this class of field to ease the following development.

6.1.3.1. *Coefficient ℓ_2-norm.* Suppose now that $\mathbf{L}$ is a cyclotomic field and take ζ a primitive root of unity so that $\mathbf{L} \cong \mathbf{Q}(\zeta)$, then $\mathscr{O}_\mathbf{L} \cong \mathbf{Z}[\zeta]$. In the *integral power basis* $(1, \zeta, \zeta^2, \cdots, \zeta^{n-1})$ any algebraic integer $x \in \mathscr{O}_\mathbf{L}$ decomposes as $\sum_{i=0}^{n-1} x_i \zeta^i$, with $x_i \in \mathbf{Z}$. This coordinate identification ι is an explicit isomorphism of $\mathbf{Z}$-modules between $\mathscr{O}_\mathbf{L}$ and $\mathbf{Z}^n$ as we have $\mathscr{O}_\mathbf{L} \cong \mathbf{Z}[\zeta]$. We can then *lift* the usual norm on $\mathbf{Z}^n$ to $\mathscr{O}_\mathbf{L}$ in order to make the coefficient identification an isometry:

$$\|x\|^2 = \sum_{i=0}^{n-1} x_i^2.$$

6.1.3.2. *On the lattice structure involved by the coefficient ℓ_2-norm.* Since the ℓ_2 norm over $\mathbf{Z}^n$ is an Euclidean norm, so is its lift over $\mathscr{O}_\mathbf{L}$, making $\mathscr{O}_\mathbf{L}$, and more generally any ideal $\mathfrak{a} \subseteq \mathscr{O}_\mathbf{L}$ an Euclidean lattice. The isomorphism of $\mathbf{Z}$-modules ι makes the computation of its covolume straightforward:

Proposition 6.1.1. *Let $\mathbf{L}$ be a cyclotomic field and $\mathfrak{a}$ an ideal of $\mathscr{O}_\mathbf{L}$. Then the covolume $\mathrm{covol}_{\ell_2}(\mathfrak{a})$ of the direct image over $\mathbf{Z}$ of the $\mathscr{O}_\mathbf{L}$-lattice given by $\mathfrak{a}$ for the coefficient ℓ_2-norm for the integral power basis, is $N_\mathbf{L}(\mathfrak{a})$.*

Proof. Clearly we have:

$$\mathrm{covol}_{\ell_2}(\mathfrak{a}) = \mathrm{covol}(\iota(\mathfrak{a})) = \left|\mathbf{Z}^n/\iota(\mathfrak{a})\right| = \left|\iota(\mathscr{O}_\mathbf{L})/\iota(\mathfrak{a})\right| = \left|\mathscr{O}_\mathbf{L}/\mathfrak{a}\right| = N(\mathfrak{a}),$$

as ι is an isomorphism of $\mathbf{Z}$-modules. ∎

This should be compared to the covolume of the direct image of $\mathfrak{a}$ for the canonical norm of $\mathbf{L}$, which is $N(\mathfrak{a})\sqrt{|\Delta_\mathbf{L}|}$, by definition of the covolume given in Section 3.3.

Example. *Let us take $\mathbf{L} = \mathbf{Q}[\zeta_3]$, the cyclotomic field of conductor 3 and degree 2. Its discriminant is -3. Then, $\mathscr{O}_\mathbf{L} = \mathbf{Z}[\zeta_3]$ admits $\mathcal{B} = (1, \zeta_3)$ as a $\mathbf{Z}$-basis. The canonical embeddings are given by the identity map and by the morphism mapping ζ_3 to $\zeta_3^2 = -1 - \zeta_3$, so that the canonical norm of a generic element is $N_{\mathbf{L}/\mathbf{Q}}(x + y\zeta_3) = x^2 + y^2 - xy$. Let us construct the ideal $\mathfrak{a}$ generated by the element $(1 - \zeta_3)$, of norm equal to 3. The corresponding $\mathbf{Z}$-module is generated by $(3, 1 - \zeta_3)$ Hence under the ℓ_2 coefficient norm in $\mathcal{B}$, the corresponding lattice is isometric to :*

$$\Lambda_{\ell_2} = \left(\mathbf{Z}^2, \begin{pmatrix} 9 & 3 \\ 3 & 2 \end{pmatrix}\right).$$

Indeed, the coefficients of 3 is of course $(3, 0)^T$, which is a vector of norm 3, the coefficients of $1 - \zeta_3$ are $(1, 1)^T$ of norm $\sqrt{2}$. The inner product of these two vectors is $3 \times 1 + 0 \times 1 = 3$.

Under the canonical norm it is isometric to:

$$\Lambda_c = \left(\mathbf{Z}^2, \begin{pmatrix} 18 & 9 \\ 9 & 4 \end{pmatrix}\right).$$

We can check that $|\mathrm{covol}(\Lambda_c)/\mathrm{covol}(\Lambda_{\ell_2})|^2 = 3 = |\Delta_\mathbf{L}|$. Indeed, the embedding of 3 is the vector $(3, 3)^T$ of norm $3\sqrt{2}$, the embedding of $1 - \zeta_3$ is $(1 - \zeta_3, 2 + \zeta_3)^T$ of norm 4 and the inner product of these two embeddings is $3(1 - \bar{\zeta}_3) + 3(2 + \bar{\zeta}_3) = 9$.

The Figure 1 and Figure 2 depict the difference between these two embeddings.

We have the same result for the maximal real subfield $\mathbf{L}^+$ of $\mathbf{L}$, this time for the *folded power basis*. Define ς^+ the coefficient embedding for the basis $(\zeta^i + \zeta^{-i})_i$ of $\mathscr{O}_{\mathbf{L}^+}$ (see Proposition 1.6.3 for the structure result of this ring).

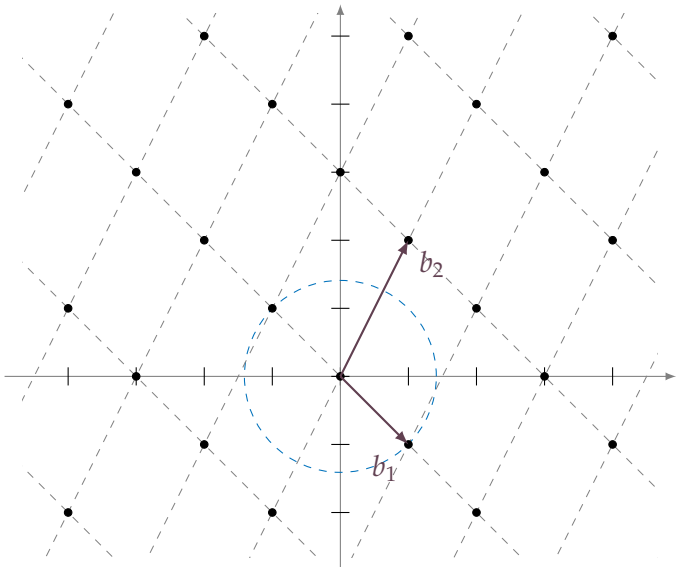


Figure 1: Lattice generated by $(1-\zeta_3) \subset \mathbf{Z}[\zeta_3]$ through the coefficient embedding.

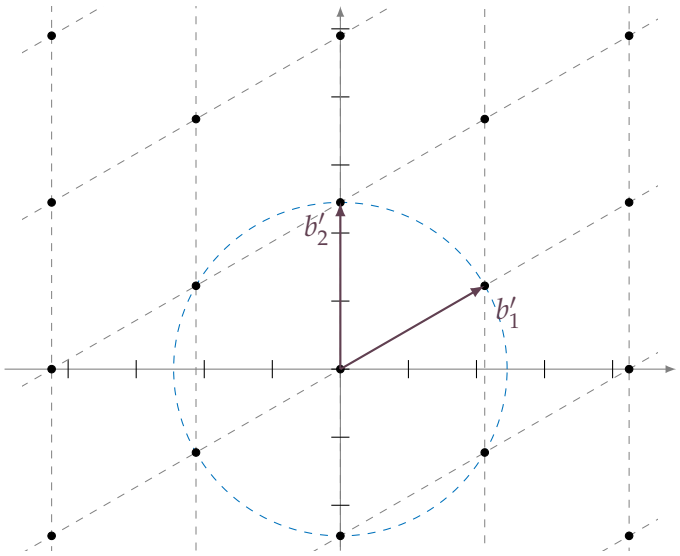


Figure 2: Lattice generated by $(1-\zeta_3) \subset \mathbf{Z}[\zeta_3]$ through the canonical embedding.

Proposition 6.1.2. *Let* $\mathbf{L}$ *be a cyclotomic field,* $\mathbf{L}^+$ *its maximal real subfield, and* $\mathfrak{a}$ *an ideal of* $\mathcal{O}_{\mathbf{L}^+}$*. Then the covolume* $\mathrm{covol}_{\ell_2}(\mathfrak{a})$ *of the direct image over* $\mathbf{Z}$ *of the* $\mathcal{O}_{\mathbf{L}^+}$*-lattice given by* $\mathfrak{a}$ *for the embedding* ς^+*, is* $N_{\mathbf{L}^+}(\mathfrak{a})$*.*

Proof. The proof is done in the exact same way as for Proposition 6.1.1, using this time the fact that $\varsigma^+(\mathcal{O}_{\mathbf{L}^+}) = \mathbf{Z}^{\frac{n}{2}}$ as $\mathcal{O}_{\mathbf{L}^+} = \mathbf{Z}[\zeta + \zeta^{-1}]$ from Proposition 1.6.3. ∎

This should be compared to the covolume of the direct image of $\mathfrak{a}$ for the canonical norm of $\mathbf{L}$, which gives: $\mathrm{covol}(\mathfrak{a}) = N(\mathfrak{a})\sqrt{|\Delta_{\mathbf{L}}|}$.

6.2 SOLVING THE PRINCIPAL IDEAL PROBLEM

6.2.1 Setting

In all of the following let us fix a cyclotomic field $\mathbf{L} = \mathbf{Q}[\zeta_f]$ of conductor f and degree $n = \varphi(f)$. For the sake of simplicity, we now write ζ for ζ_f. The principal ideal problem is formally stated as follow:

Problem (Principal ideal problem the number field $\mathbf{L}$)**.** *Let* $g \in \mathcal{O}_{\mathbf{L}}$ *an integer. Denote by* $\mathfrak{a} = g\mathcal{O}_{\mathbf{L}}$ *the ideal generated by this element.*

> ***Input:*** *$\mathcal{B}$, a basis of $\mathfrak{a}$ as a free $\mathbf{Z}-$module of rank n.*
>
> ***Output:*** *The generator $g \in \mathcal{O}_{\mathbf{L}}$, modulo the unit group $\mathcal{O}_{\mathbf{L}}^{\times}$.*

We first give an overview of the whole strategy we use to solve this problem.

6.2.2 High level description

Before any other operations, the ambient dimension of the problem is shrunk by half by reducing the problem to an equivalent one in the *maximal real subfield* $\mathbf{L}^+ = \mathbf{Q}(\zeta + \zeta^{-1})$. This reduction is not necessary from a theoretical point of view, but eases the computation as we are working with objects of smaller dimension, and decrease the asymptotic complexity. This reduction is a straightforward consequence of the Gentry-Szydlo algorithm introduced in Section 5.6.1. Hence the problem is now reduced to the search for a generator of a principal ideal $\mathfrak{a}^+$ in $\mathbf{L}^+$. The strategy then splits in three steps.

First, iteratively reduce the size of the ideal whereof we look for a generator—while staying in the class of principal ideals—until we eventually reach a B-smooth ideal for a fixed bound $B > 0$. Formally this breaks down to the construction of an algebraic integer h and a B-smooth ideal $\mathfrak{a}^s$, such that $h\mathcal{O}_{\mathbf{L}} = \mathfrak{a}^+ \cdot \mathfrak{a}^s$. This is the *descent phase*. The process is illustrated in Figure 3.

The next step consists in finding a generator of $\mathfrak{a}^s$. We use a strategy based on class group computation. It boils down to constructing a generating set

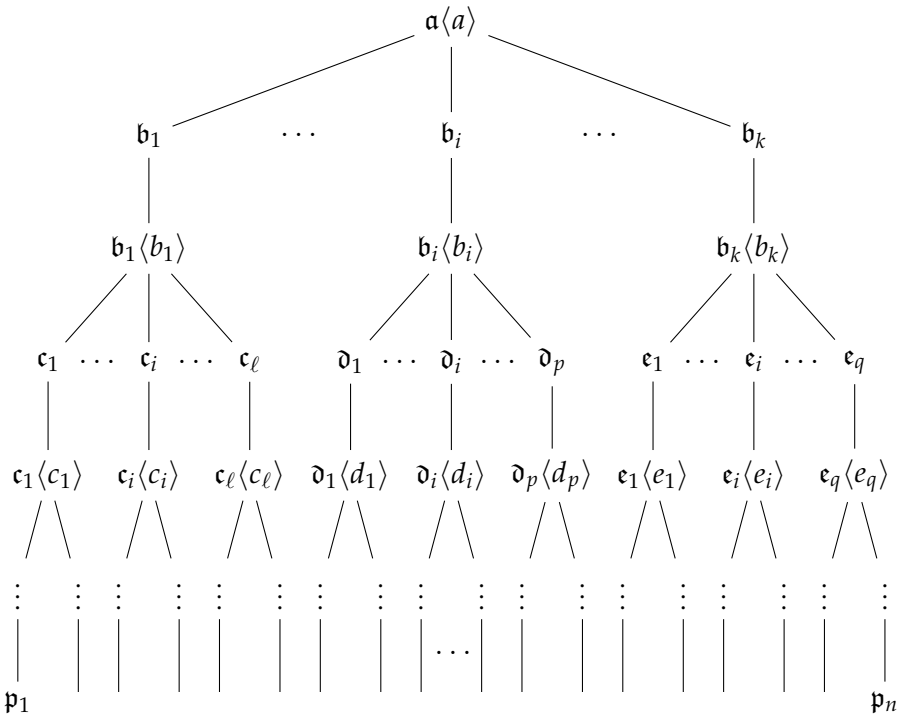


Figure 3: Description of the descent to small norm ideals. The smoothness of the ideals is decreasing along the descent tree. The leaves are all B-smooth.

of all the relations among the small generators of the class group, and then rewrite the input ideal with respect to these generators. Then we can recover a generator h_0 of $\mathfrak{a}^s$ by simply solving a linear system of equations.

The latter allows us to derive a generator of the ideal $\mathfrak{a}^+ \colon h \cdot {h_0}^{-1}$. A generator of the initial ideal is eventually obtained by lifting a generator of the ideal $\mathfrak{a}^+$ from the maximal real subfield to the initial number field $\mathbf{L}$; this final step being actually a multiplication by an algebraic integer.

In a nutshell, the full algorithm splits in four main steps, which are :

1. Perform a reduction from the cyclotomic field $\mathbf{L}$ to its maximal real subfield $\mathbf{L}^+$, allowing to work in smaller dimension.
2. Iteratively lowers the norm of the ideals involved.
3. Collect relations on small prime ideals and run linear algebra to reconstruct the generator of the small ideal generated at step 2.
4. Lift the generator from $\mathbf{L}^+$ to $\mathbf{L}$.

The remaining part of this chapter is devoted to prove the following complexity theorem:

Theorem 6.2.1. *Let $\mathbf{L} = \mathbf{Q}(\zeta_f)$, which discriminant is denoted by Δ. There exists a heuristic Las-Vegas algorithm which solves the principal ideal problem in $\mathbf{L}$ in expected time:*

$$L_\Delta\left[\frac{1}{2}, \frac{\omega}{2\sqrt{2(\omega-1)}}\right].$$

Let us now dive into the details of all steps sketched above in order to prove Theorem 6.2.1. The estimate of the second constant $\frac{\omega}{2\sqrt{2(\omega-1)}}$ of the complexity is only addressed at the end of Section 6.3.2 in order to lighten the algorithm description and ease the presentation, so that we only aim at ensuring a complexity of $L_\Delta\left[\frac{1}{2}\right]$ in a first time.

Recall that we work in $\mathbf{L} = \mathbf{Q}[\zeta]$ of conductor f and degree $n = \varphi(f)$. For any element $u \in \mathbf{L}$, we denote by $\bar{u}$ the conjugate of u for the automorphism defined by $\zeta \mapsto \zeta^{-1}$. We are given a principal ideal $\mathfrak{a} = g\mathcal{O}_\mathbf{L}$ represented by a $\mathbf{Z}$-basis $\mathcal{B} = (b_1, \ldots, b_n)$.

6.2.3 Step 1: Reduction to the maximal real subfield

As usual when doing computations, we are subjected to the *curse of dimensionality*: the larger the dimension becomes, the more costly the algebraic numbers are to handle and even only to represent, impacting the efficiency of any algorithm. However, in the present setting it is possible to halve the dimension in which the problem occurs with a polynomial time reduction. The main part of this step relies on the so-called *Gentry-Szydlo* algorithm, presented in Section 5.6.1.

6.2.3.1. *Reduction to the Gentry-Szydlo setting.* This original algorithm from [64] takes as input a $\mathbf{Z}$-basis of a principal ideal $\mathfrak{b}$ in the ring $\mathcal{O}_\mathbf{L}$ and an algebraic integer of the form $b \cdot \bar{b}$, for b a generator of $\mathfrak{b}$. It then recovers in polynomial time the element b. In our case, we cannot perform the recovery of the generator g of the input ideal since *a priori* we do not have access to any kind of information about the product $g \cdot \bar{g}$.

To overcome this difficulty, let us introduce another algebraic integer

$$u = N_{\mathbf{L}/\mathbf{Q}}(g)\frac{g}{\bar{g}},$$

as described by Garg, Gentry, and Halevi in [58, Section 7.8.1]. Here the norm factor is included only to avoid the introduction of denominators in the definition of u. Although u is still unknown at this point, thanks to the $\mathbf{Z}$-basis of $g\mathcal{O}_\mathbf{L} = \mathfrak{a}$ we can construct a $\mathbf{Z}$-basis of $u\mathcal{O}_\mathbf{L}$, since we have:

$$u\mathcal{O}_\mathbf{L} = N_{\mathbf{L}/\mathbf{Q}}(g)\frac{g}{\bar{g}}\mathcal{O}_\mathbf{L} = N(\mathfrak{a})\mathfrak{a}\overline{\mathfrak{a}^{-1}}.$$

Moreover, we can compute the product $u \cdot \bar{u}$, as it simply corresponds to $N_{\mathbf{L}/\mathbf{Q}}(g)^2 = N(\mathfrak{a})^2$.

From this point we can compute u in polynomial time using the *Gentry-Szydlo* algorithm, and from this element u, we can directly recover

$$\frac{g}{\bar{g}} = uN(\mathfrak{a})^{-1}.$$

Using the basis $\mathcal{B}$ of $\mathfrak{a}$, we then introduce the family of vectors

$$c_i = b_i\left(1 + \frac{\bar{g}}{g}\right),$$

which is a basis of the ideal $\mathfrak{a}^+$ generated by $g + \bar{g}$. This ideal belongs to the maximal real subfield $\mathbf{L}^+ = \mathbf{Q}(\zeta + \zeta^{-1})$, of index 2 in $\mathbf{L}$.

6.2.3.2. *Lifting the maximal real solution.* Suppose that we know the generator $(g + \bar{g})$ of $\mathfrak{a}^+$, up to a unit. Then remark that:

$$(g + \bar{g})\left(\frac{1}{1 + g\,\bar{g}^{-1}}\right) = \frac{\bar{g}(g + \bar{g})}{\bar{g} + g} = \bar{g},$$

so that we can recover the generator g. Hence, we have reduced the problem of finding a generator of the ideal $\mathfrak{a}$ belonging to the cyclotomic field $\mathbf{L}$ of dimension n to the one of finding a generator of ideal $\mathfrak{a}^+$ that belongs to the maximal real subfield $\mathbf{L}^+$, whose degree over $\mathbf{Q}$ is $\frac{n}{2}$.

6.2.4 Step 2: Descent phase

Let us set aside the algebraic integer obtained in the previous phase and only focus on the ideal $\mathfrak{a}^+$. By construction, it is principal and generated by $g + \bar{g}$, so that we aim at retrieving this generator by solving the principal ideal problem in the maximal real subfield $\mathbf{L}^+$ for $\mathfrak{a}^+$.

6.2.4.1. *Reducing the general problem to the PIP for small ideals.* Suppose that we can solve efficiently this problem for all principal ideals which are B-smooth, for a certain bound B, which is fixed. In this context, solving PIP in its whole generality could be done by reduction to the B-smooth case. Formally this translates in the construction of an algebraic integer h and a B-smooth principal ideal $\mathfrak{a}^s$, such that $h\mathcal{O}_{\mathbf{L}} = \mathfrak{a}^+ \cdot \mathfrak{a}^s$.

6.2.4.2. *Construction of the elements.* The element h and ideal $\mathfrak{a}^s$ are constructed iteratively, by generating at each step ideals of norm smaller and smaller until eventually finding a B-smooth one. We demonstrate in Paragraph 6.2.4.4 that this descent is doable in expected time $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$ from any ideal of algebraic norm bounded in $\mathrm{L}_\Delta[1]$.

However the initial ideal $\mathfrak{a}^+$ is of arbitrary norm. As such in order to bootstrap this phase, we first need to find an ideal that splits as a product of prime ideals of controlled norm on which we will independently apply the descent. In this context being able to bounded the norm of these ideals $\mathrm{L}_\Delta[1]$ suffices.

6.2.4.3. *Initial round* Hence we aim at constructing efficiently an $\mathrm{L}_\Delta[1]$-smooth principal ideal from $\mathfrak{a}^+$. Formally, we want to prove:

Theorem 6.2.2. *Let $\mathbf{L} = \mathbf{Q}[\zeta_f]$ the cyclotomic field of conductor f and $\mathbf{L}^+$ its maximal real subfield. Denote by Δ its discriminant. Assuming Heuristic 6.1.1, from any ideal $\mathfrak{a} \subset \mathcal{O}_{\mathbf{L}^+}$, there exists a probabilistic algorithm which generates in expected time $L_\Delta\left[\frac{1}{2}\right]$ an integral ideal $\mathfrak{b}$ that is $L_\Delta[1]$-smooth and an algebraic integer v such that*

$$v\mathcal{O}_{\mathbf{L}} = \mathfrak{a} \cdot \mathfrak{b}.$$

The difficulty of this preliminary part is that *a priori* the norm of the input ideal $\mathfrak{a}$ can be large. We thus want to construct at first a candidate ideal $\mathfrak{a}'$ whose norm is bounded *independently* from $N_{\mathbf{L}^+}(\mathfrak{a})$ and that belongs to the same ideal class as $\mathfrak{a}$. The probabilistic argument of Heuristic 6.1.1 states that if this norm is small enough then the probability of being smooth will be large enough to be easily amplified by repetition.

Lemma 6.2.1. *Let $\mathbf{L} = \mathbf{Q}[\zeta_f]$ the cyclotomic field of conductor f and degree $n = \varphi(f)$, and $\mathbf{L}^+$ its maximal real subfield. Denote by Δ the discriminant of $\mathbf{L}^+$. Assuming Heuristic 6.1.1, from any ideal $\mathfrak{a} \subset \mathscr{O}_{\mathbf{L}}$, it is possible to generate in deterministic time $L_\Delta\left[\frac{1}{2}\right]$ an integral ideal $\mathfrak{b}$ of norm bounded by $L_\Delta\left[\frac{3}{2}\right]$ and an algebraic integer v such that*

$$v\mathscr{O}_{\mathbf{L}^+} = \mathfrak{a} \cdot \mathfrak{b}.$$

Proof. We proceed by lattice reduction, since $\mathfrak{a}$ is endowed with a natural lattice structure from the canonical norm (see Section 6.1.3). The covolume of the direct image of the lattice $\mathfrak{a}$ over $\mathbf{Z}$ is $\sqrt{\Delta} \cdot N_{\mathbf{L}^+}(\mathfrak{a})$.

Given an integer $2 \leq \beta \leq \frac{n}{2}$, it follows from Theorem 2.5.1 on the quantitative analysis of the DBKZ algorithm that the norm of the smallest vector v of a DBKZ_β−reduced basis of $\mathfrak{a}$ satisfies:

$$\|v\|_{\mathbf{L}^+} \leq \beta^{\frac{n/2-1}{2(\beta-1)}} N_{\mathbf{L}^+}(\mathfrak{a})^{\frac{2}{n}} \Delta^{\frac{1}{n}}, \tag{6.2}$$

the cost of this reduction being upper bounded by $\text{Poly}(n, \log N_{\mathbf{L}^+}(\mathfrak{a}))2^{O(\beta)}$.

Since the ideal $\mathfrak{a}$ contains $v\mathscr{O}_{\mathbf{L}^+}$, there exists a unique integral ideal $\mathfrak{b}$ satisfying $v\mathscr{O}_{\mathbf{L}^+} = \mathfrak{a} \cdot \mathfrak{b}$. From Equation 6.2, we have:

$$N_{\mathbf{L}^+/\mathbf{Q}}(v\mathscr{O}_{\mathbf{L}^+}) \leq \beta^{\frac{n/2(n/2-1)}{2(\beta-1)}} \cdot \sqrt{\Delta} \cdot N_{\mathbf{L}^+}(\mathfrak{a}).$$

and by the multiplicative property of the norm, we find: in

$$N_{\mathbf{L}^+}(\mathfrak{b}) \leq \beta^{\frac{n(n-2)}{8(\beta-1)}} \cdot \sqrt{\Delta}.$$

Since $\mathbf{L}$ is a cyclotomic field, we are able to choose a block-size $\beta = \log\left(L_\Delta\left[\frac{1}{2}\right]\right)$ since $\log\left(L_\Delta\left[\frac{1}{2}\right]\right) = n^{\frac{1}{2}+o(1)} \leq n$ by the remark of Section 6.1.1. Then we are able to generate in time

$$\text{Poly}(n, \log N_{\mathbf{L}^+}(\mathfrak{a}))2^{O(\beta)} = \text{Poly}(\log N_{\mathbf{L}^+}(\mathfrak{a}))L_\Delta\left[\frac{1}{2}\right]$$

an integral ideal of norm bounded by

$$\beta^{\frac{n(n-2)}{8(\beta-1)}} \cdot \sqrt{\Delta} = \log\left(L_\Delta\left[\frac{1}{2}\right]\right)^{\frac{n^2}{\log L_\Delta\left[\frac{1}{2}\right]}} = \exp\left[\log\log n^{\frac{1}{2}+o(1)} n^{\frac{3}{2}+o(1)}\right] = L_\Delta\left[\frac{3}{2}\right].$$

■

This last result allows us to find an ideal whose norm is bounded independently from $N_{\mathbf{L}}(\mathfrak{a})$. We then want this new ideal to split as a product of multiple prime ideals of controlled norms. Thanks to Corollary 6.1.1, the probability for an integral ideal $\mathfrak{b}$ of norm bounded by $\mathrm{L}_\Delta\left[\frac{3}{2}\right]$ to be $\mathrm{L}_\Delta[1]$-smooth is greater than $\mathrm{L}_\Delta\left[\frac{1}{2}\right]^{-1}$, so that a randomized version of the latter construction will yield a smooth ideal after $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$ repetitions. This is the crux of the following proof.

Proof of Theorem 6.2.2. The simplest strategy to perform this randomization and be able to repeat the construction, is to compose the input ideal $\mathfrak{a}$ with some factors of norm less than $B = \mathrm{L}_\Delta\left[\frac{1}{2}\right]$. Formally, we denote by $\mathcal{B} = \{\mathfrak{p}_1, \dots, \mathfrak{p}_{|\mathcal{B}|}\}$ the set of all prime ideals of norm upper bounded by $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$. As a consequence of Landau's Prime Ideal Theorem [103],

$$|\mathcal{B}| \sim \mathrm{L}_\Delta\left[\frac{1}{2}\right]\left(\log \mathrm{L}_\Delta\left[\frac{1}{2}\right]\right)^{-1} = \mathrm{L}_\Delta\left[\frac{1}{2}\right].$$

Finding all these ideals can be done in a very naive way by factoring each of the $\mathrm{O}\left(\frac{B}{\log B}\right)$ prime integers $1 < p < B$ into primes ideals using polynomial factorization over finite fields, which yields a running time bounded by $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$.

Let $k, A > 0$ be fixed integers. We choose $\mathfrak{p}_{j_1}, \dots, \mathfrak{p}_{j_k}$ prime ideals of norm below $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$. Then for any k-tuple $e = (e_1, \dots, e_k) \in \{1, \dots, A\}^k$, and a k-tuple $(j_1, \dots, j_k) \subset \{1, \dots, |\mathcal{B}|\}$, denote by $\mathfrak{P}_e = \mathfrak{p}_{j_1}^{e_1} \cdots \mathfrak{p}_{j_k}^{e_k}$ so that we have:

$$\begin{aligned} N_{\mathbf{L}^+}(\mathfrak{a}\mathfrak{P}_e) &= N_{\mathbf{L}^+}\left(\mathfrak{a} \cdot \prod_{i=1}^{k} \mathfrak{p}_{j_i}^{e_i}\right) \\ &\leq N_{\mathbf{L}^+}(\mathfrak{a}) \cdot \prod_{i=1}^{k} N_{\mathbf{L}}\left(\mathfrak{p}_{j_i}^{e_i}\right) \leq N_{\mathbf{L}^+}(\mathfrak{a}) \cdot \mathrm{L}_\Delta\left[\frac{1}{2}\right]^{k \cdot A} \\ &= N_{\mathbf{L}^+}(\mathfrak{a}) \cdot \mathrm{L}_\Delta\left[\frac{1}{2}\right]. \end{aligned}$$

Therefore the randomization can be done by choosing uniformly at random the tuple $(e_1, \dots, e_k)$ and k prime ideals in $\mathcal{B}$. Since $|\mathcal{B}| = \mathrm{L}_\Delta\left[\frac{1}{2}\right]$, the set of possible samples is large enough for sampling a $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$ *independent* number of different ideals.

Using ECM to test for smoothness yields a complexity in $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$ (see Equation 6.1). Therefore, we can repeat the construction sketched in the proof of Lemma 6.2.1, on randomized independent inputs of the shape $\mathfrak{a}\mathfrak{P}_e$ until one of them eventually yields an $\mathrm{L}_\Delta[1]$-smooth ideal $\tilde{\mathfrak{b}}$. By Heuristic 6.1.1, the expected number of repetitions is a $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$. To conclude it suffices to remark that the ideal $\tilde{\mathfrak{b}} \cdot \mathfrak{P}_e$ is of course $\mathrm{L}_\Delta[1]-$smooth. ∎

Other ways to perform the randomization may be by randomizing directly the lattice reduction algorithm or by enumerating points of the lattice of norm close to the norm guarantee and change the basis vectors by freshly

enumerated ones. The latter would be useful in practice as it reduces the number of reductions. However the asymptotic analysis would remain unchanged.

The full outline of this bootstrapping approach, illustrated in the proof of Theorem 6.2.2, is given in pseudocode as Algorithm 25, **Bootstrap**.

Algorithm 25 — Bootstrap

Input : An arbitrary ideal $\mathfrak{a}$ and constants B, A, and k as in the proof of Theorem 6.2.2.
Output : An integer v and an $\mathrm{L}_\Delta[1]$-smooth ideal $\mathfrak{b}$ such that $v\mathscr{O}_{\mathbf{L}^+} = \mathfrak{a}\mathfrak{b}$

// Precomputation of $\mathcal{B}$, independant of $\mathfrak{a}$
$\mathcal{B} = \emptyset$
$\mathcal{P}_{\leq B} \leftarrow$ sieve primes below B
for $p \in \mathcal{P}_{\leq B}$ **do**
  Factor $p\mathscr{O}_{\mathbf{L}^+}$ in $\prod_i \mathfrak{p}_i^{e_i}$
  $\mathfrak{P}_p = \{\mathfrak{p}_i\}_{\mathfrak{p}_i \mid (p)}$
  $\mathcal{B} \leftarrow \mathcal{B} \cup \mathfrak{P}_p$
end for
// Bootstraping phase of $\mathfrak{a}$
$\mathfrak{b} \leftarrow \mathfrak{a}$
while $\mathfrak{b}$ *is not* $L_\Delta[1]$*-smooth* **do**
  Choose $\mathfrak{p}_{j_1}, \dots, \mathfrak{p}_{j_k}$ uniformly at random in $\mathcal{B}$
  $\mathfrak{P}_e \leftarrow \mathfrak{a} \cdot \prod \mathfrak{p}_{j_i}^{e_i}$ for random $e_i \in \{0, \dots, A\}$
  $\tilde{\mathfrak{a}} \leftarrow \mathfrak{a} \cdot \mathfrak{P}_e$ // Computed using $\mathbf{Z}$-bases
  $v \leftarrow$ shortest vector of $\mathbf{DBKZ}((\mathfrak{a}, \|\cdot\|_{\mathbf{L}^+})_*)$ // See Chapter 3 for the definition of the direct image
  $\tilde{\mathfrak{b}} \leftarrow v\tilde{\mathfrak{a}}^{-1}$ // Computed with $\mathbf{Z}$-bases
  $\mathfrak{b} \leftarrow \tilde{\mathfrak{b}} \cdot \prod_{i=1}^{k} \mathfrak{p}_{j_i}^{-e_i}$
end while
return $v, \mathfrak{b}$

6.2.4.4. *From $L_\Delta[1]$ to $L_\Delta\left[\frac{1}{2}\right]$—smooth ideals.* In the proof of Theorem 6.2.2, we used directly the bound obtained from the DBKZ reduction. We could not use Corollary 2.6.1 on the reduction of a part of the HNFof the basis here, since the norm of the ideal $\mathfrak{a}^+$ and thus its volume can be arbitrary large. Nonetheless, the norm of prime ideals appearing in its factorization are by the smoothness condition bounded, making possible the use of this pre-treatment of the underlying lattice. The systematic treatment of this question is the aim of Theorem 6.2.3.

Theorem 6.2.3. *Let $\mathbf{L} = \mathbf{Q}[\zeta_f]$ the cyclotomic field of conductor f and degree $n = \varphi(f)$, and $\mathbf{L}^+$ its maximal real subfield. Denote by Δ the discriminant of $\mathbf{L}^+$. Let $\mathfrak{a}$ be an ideal of $\mathbf{L}^+$ of norm below $L_\Delta[\alpha]$, for $\frac{1}{2} \leq \alpha \leq 1$. Then, in*

expected time $L_\Delta\left[\frac{1}{2}\right]$*, it is possible to construct an algebraic integer* v *and an* $L_\Delta\left[\frac{2\alpha+1}{4}\right]$*-smooth ideal* $\mathfrak{b}$ *such that*

$$v\mathcal{O}_{\mathbf{L}^+} = \mathfrak{a}\cdot\mathfrak{b}.$$

The core of the proof is similar to the proof of Theorem 6.2.2 as it relies on lattice reduction backed up with randomization. The difference lies here in the norm with respect to which the reduction is performed. In Theorem 6.2.2, the canonical norm is used, whereas here we use directly the structure arising from the coefficient embedding ς. In the present context it allows a better bound on the size of the short vector found by reduction and therefore a better bound on the norm of the resulting ideal $\mathfrak{b}$.

Proof of Theorem 6.2.3. Remark that since we work in the maximal real subfield, we can use the folded coefficient embedding ς^+ for the $\mathbf{Z}$-basis $(\zeta^i + \zeta^{-i})_i$, defined in Proposition 6.1.2. Hence, we have for $v \in \mathcal{O}_{\mathbf{L}}^+$,

$$\|\varsigma(v)\|_2 = \sqrt{2}\|\varsigma^+(v)\|_2,$$

where the $\|\cdot\|_2$ is the canonical ℓ_2 norm over the spaces $\mathbf{Q}^{\frac{n}{2}}$ for the right hand side and $\mathbf{Q}^n$ for the left hand side.

Let $\|\cdot\|_{\varsigma^+}$ the corresponding norm on $\mathfrak{a}$, yielding an $\mathcal{O}_{\mathbf{L}^+}$-lattice $\left(\mathfrak{a}, \|\cdot\|_{\varsigma^+}\right)$. The covolume of its direct image Λ over $\mathbf{Z}$ is then $N_{\mathbf{L}^+}(\mathfrak{a})$. Then, using the block-size $\beta = \log \mathrm{L}_\Delta\left[\frac{1}{2}\right] = n^{\frac{1}{2}+\mathrm{o}(1)}$, we have

$$\operatorname{covol}\Lambda \leq \mathrm{L}_\Delta[\alpha] = 2^{\mathrm{O}(n^\alpha \log(n))} \leq \beta^{\frac{n^2}{2\beta}}.$$

Using the DBKZ reduction algorithm within the analysis of Corollary 2.6.1 yields in time $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$ an integer v satisfying

$$\begin{aligned}\|v\|_{\varsigma^+} \leq \beta^{\sqrt{\frac{2\log_\beta(\operatorname{covol}\Lambda)}{\beta}}(1+o(1))} &= \exp\left[(1+\mathrm{o}(1))\log(n)\sqrt{n^{\alpha-\frac{1}{2}+\mathrm{o}(1)}}\right] \\ &= \mathrm{L}_\Delta\left[\frac{\alpha}{2}-\frac{1}{4}\right].\end{aligned}$$

Formulating this bound in terms of the field norm induces:

$$\begin{aligned}N_{\mathbf{L}^+}(v) \leq \left(\sqrt{2}\left(\frac{n}{2}+1\right)\right)^{\frac{n}{2}} \cdot \|v\|_{\varsigma^+}^{\frac{n}{2}} &= \mathrm{L}_\Delta[1]\cdot\left(\mathrm{L}_\Delta\left[\frac{\alpha}{2}-\frac{1}{4}\right]\right)^{\frac{n}{2}} \\ &= \mathrm{L}_\Delta[1]\cdot\mathrm{L}_\Delta\left[\frac{\alpha}{2}+\frac{3}{4}\right].\end{aligned}$$

Since $\alpha \geq \frac{1}{2}$, we then have $N_{\mathbf{L}^+}(v\mathcal{O}_{\mathbf{L}^+}) = N_{\mathbf{L}^+/\mathbf{Q}}(v) = \mathrm{L}_\Delta\left[\frac{\alpha}{2}+\frac{3}{4}\right]$.

Because the ideal $\mathfrak{a}$ contains $v\mathcal{O}_{\mathbf{L}^+}$, there exists a unique integral ideal $\mathfrak{b}$, satisfying $v\mathcal{O}_{\mathbf{L}^+} = \mathfrak{a}\cdot\mathfrak{b}$. We find that $N_{\mathbf{L}^+}(\mathfrak{b}) \leq \mathrm{L}_\Delta\left[\frac{\alpha}{2}+\frac{3}{4}\right]$ from the multiplicative property of the norm and $N_{\mathbf{L}^+}(\mathfrak{a}) = \mathrm{L}_\Delta[1] \leq \mathrm{L}_\Delta\left[\frac{\alpha}{2}+\frac{3}{4}\right]$. Under Heuristic 6.1.1, this ideal is $\mathrm{L}_\Delta\left[\frac{\alpha}{2}+\frac{1}{4}\right]$-smooth with probability $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$. Eventually performing the randomization-and-repeat technique as in Algorithm 25, the reduction yields the desired couple $(v, \mathfrak{b})$ in expected time $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$. ∎

6.2.5 Putting both parts together.

Thanks to Theorem 6.2.2 we can generate a $L_\Delta[1]$-smooth ideal, denoted by $\mathfrak{a}^{(0)}$, and an algebraic integer $h^{(0)}$ satisfying

$$h^{(0)}\mathcal{O}_{\mathbf{L}^+} = \mathfrak{a}^+ \cdot \mathfrak{a}^{(0)},$$

with $\mathfrak{a}^+$ the ideal of the maximal real subfield obtained as a result of the computation of Section 6.2.3. The factorization of $\mathfrak{a}^{(0)}$ gives

$$\mathfrak{a}^{(0)} = \prod_j \mathfrak{a}_j^{(0)},$$

where the $\mathfrak{a}_j^{(0)}$ are integral prime ideals of norm upper bounded by $L_\Delta[1]$. Remark that we do not need to perform the descent on the $\mathfrak{a}_j^{(0)}$ which are *already* of norm below $B = L_\Delta\left[\frac{1}{2}\right]$, so that we only continue the process for the subset

$$S = \left\{\mathfrak{a}_j^{(0)} \mid N\left(\mathfrak{a}_j^{(0)}\right) > L_\Delta\left[\frac{1}{2}\right]\right\}.$$

Taking the norms of these ideals yields:

$$N\left(\mathfrak{a}^{(0)}\right) \geq \prod_{\mathfrak{a}_j^{(0)} \in S} N\left(\mathfrak{a}_j^{(0)}\right),$$

yielding the inequality $\log L_\Delta\left[\frac{3}{2}\right] \geq |S| \log L_\Delta\left[\frac{1}{2}\right]$. Thus, $|S| = O(n_\mathcal{I})$, with $n_\mathcal{I} = \frac{\log \Delta}{\log\log \Delta} = O(n)$. Then applying Theorem 6.2.3 to each small ideal $\mathfrak{a}_j^{(0)}$ gives rise in expected time $L_\Delta\left[\frac{1}{2}\right]$ to ideals $\mathfrak{a}_j^{(1)}$ that are $L_\Delta\left[\frac{2\times 1+1}{4}\right] = L_\Delta\left[\frac{3}{4}\right]$-smooth and integers $h_j^{(1)}$ such that for every j,

$$h_j^{(1)}\mathcal{O}_{\mathbf{L}^+} = \mathfrak{a}_j^{(0)} \cdot \mathfrak{a}_j^{(1)}.$$

For each factor $\mathfrak{a}_j^{(1)}$, let us write its prime decomposition:

$$\mathfrak{a}_j^{(1)} = \prod_k \mathfrak{a}_{j,k}^{(1)}.$$

Once again, the number of terms appearing is a $O(n_\mathcal{I})$. Since we have $N_{\mathbf{L}^+}\left(\mathfrak{a}_{j,k}^{(1)}\right) = L_\Delta\left[\frac{3}{4}\right]$, performing the same procedure for each ideal $\mathfrak{a}_{j,k}^{(1)}$ then yields $L_\Delta\left[\frac{5}{8}\right]$-smooth ideals $\mathfrak{a}_{j,k}^{(2)}$ and integers $h_{j,k}^{(2)}$ such that

$$h_{j,k}^{(2)}\mathcal{O}_{\mathbf{L}^+} = \mathfrak{a}_{j,k}^{(1)} \cdot \mathfrak{a}_{j,k}^{(2)},$$

again in expected time $L_\Delta\left[\frac{1}{2}\right]$. Remark that this smoothness bound $L_\Delta\left[\frac{5}{8}\right]$ is obtained as $L_\Delta\left[\frac{2\times 3+1}{4}\right]$, as follows from Theorem 6.2.3.

This reasoning naturally leads to an iterative strategy for reduction. At step k, we want to reduce an ideal $\mathfrak{a}_{a_1,\ldots,a_{k-1}}^{(k-1)}$ which is $L_\Delta\left[\frac{1}{2} + \frac{1}{2^{k+1}}\right]$-smooth. As before, we have a decomposition — with $O(n_\mathcal{I})$ terms — in smaller ideals:

$$\mathfrak{a}_{a_1,\ldots,a_{k-1}}^{(k-1)} = \prod_j \mathfrak{a}_{a_1,\ldots,a_{k-1},j}^{(k-1)}.$$

Using Theorem 6.2.3 on each factor $\mathfrak{a}^{(k-1)}_{a_1,\dots,a_{k-1},j}$ whose norm is upper bounded by $L_\Delta\left[\frac{1}{2}+\frac{1}{2^{k+1}}\right]$ leads to $L_\Delta\left[\frac{1}{2}+\frac{1}{2^{k+2}}\right]$-smooth ideals $\mathfrak{a}^{(k)}_{a_1,\dots,a_{k-1},j}$ and algebraic integers $h^{(k)}_{a_1,\dots,a_{k-1},j}$ such that

$$h^{(k)}_{a_1,\dots,a_{k-1},j}\mathcal{O}_{\mathbf{L}^+} = \mathfrak{a}^{(k-1)}_{a_1,\dots,a_{k-1},j}\cdot\mathfrak{a}^{(k)}_{a_1,\dots,a_{k-1},j},$$

since $\frac{2\times\left(\frac{1}{2}+\frac{1}{2^{k+1}}\right)+1}{4} = \frac{1}{2}+\frac{1}{2^{k+2}}$.

As a consequence, one can generate $L_\Delta\left[\frac{1}{2}+\frac{1}{\log n}\right]$-smooth ideals with the previous method in *at most* $\lceil\log\log n\rceil$ steps. At this point, only $(n_{\mathcal{I}})^{\lceil\log(\log n)\rceil}$ ideals and algebraic integers appear since at each step this number is multiplied by a $O(n_{\mathcal{I}})$. As deriving a single integer/ideal pair requires expected time $L_\Delta\left[\frac{1}{2}\right]$, the overall complexity remains $L_\Delta\left[\frac{1}{2}\right]$.

6.2.5.1. *A remark on the smoothness of the bottom ideals.* Concerning the final step, a quick calculation reveals that

$$\begin{aligned}\log L_\Delta\left[\frac{1}{2}+\frac{1}{\log n}\right] &= O\left(n^{\frac{1}{2}+\frac{1}{\log n}}\log(n)\right)\\ &= O\left(n^{\frac{1}{2}}\log(n)\right)\cdot n^{\frac{1}{\log n}}.\end{aligned}$$

Since the last factor $n^{\frac{1}{\log n}}$ is $e=\exp(1)$, we obtain that

$$\log L_\Delta\left[\frac{1}{2}+\frac{1}{\log n}\right] = \log L_\Delta\left[\frac{1}{2}\right],$$

so that after at most $\lceil\log\log n\rceil$ steps, we have ideals that are $L_\Delta\left[\frac{1}{2}\right]$-smooth.

6.2.5.2. *Wrapping up and lift.* At the end of this final round, we may express the input ideal as the product of ideals for which we know a generator, and some others guaranteed to have their norm bounded by $L_\Delta\left[\frac{1}{2}\right]$. Let ℓ denote the index of the final step. To avoid having to deal with inverse ideals, we may assume without loss of generality[3] that ℓ is even. Explicitly, we have

$$\begin{aligned}h^{(0)}\mathcal{O}_{\mathbf{L}^+} &= \mathfrak{a}^+\cdot\mathfrak{a}^{(0)} = \mathfrak{a}^+\cdot\prod_{a_1}\mathfrak{a}^{(0)}_{a_1}\\ &= \mathfrak{a}^+\cdot\frac{\prod_{a_1}h^{(1)}_{a_1}\prod_{a_1,a_2,a_3}h^{(3)}_{a_1,a_2,a_3}}{\prod_{a_1,a_2}h^{(2)}_{a_1,a_2}}\mathcal{O}_{\mathbf{L}^+}\cdot\prod_{a_1,a_2,a_3}\mathfrak{a}^{(3)}_{a_1,a_2,a_3}\\ &= \mathfrak{a}^+\cdot\prod_{a_1,\dots,a_{\ell+1}}\frac{\prod_{t\in 2\mathbf{Z}+1}h^{(t)}_{a_1,\dots,a_t}}{\prod_{s\in 2\mathbf{Z}}h^{(s)}_{a_1,\dots,a_s}}\mathcal{O}_{\mathbf{L}^+}\cdot\underbrace{\prod_{a_1,\dots,a_{\ell+1}}\mathfrak{a}^{(l)}_{a_1,\dots,a_{\ell+1}}}_{:=\mathfrak{a}^s}.\end{aligned}$$

3 We can always run an additional step in the descent without changing the overall complexity.

In this last expression, the indices are chosen such that $1 \leq t \leq \ell$ and $2 \leq s \leq \ell$. We also recall that all the quantities involved belong to the maximal real subfield $\mathbf{Q}(\zeta + \zeta^{-1})$. By construction, the ideal $\mathfrak{a}^s$ is $L_\Delta\left[\frac{1}{2}\right]$-smooth and we directly get $h \in \mathscr{O}_{\mathbf{L}}^+$ such that $h\mathscr{O}_{\mathbf{L}^+} = \mathfrak{a}^+ \cdot \mathfrak{a}^s$. The full outline of this descent phase is sketched in Figure 4. Remark that the number of terms, which is at most $O(n)^\ell = L_\Delta[o(1)]$, is negligible in the final complexity estimate.

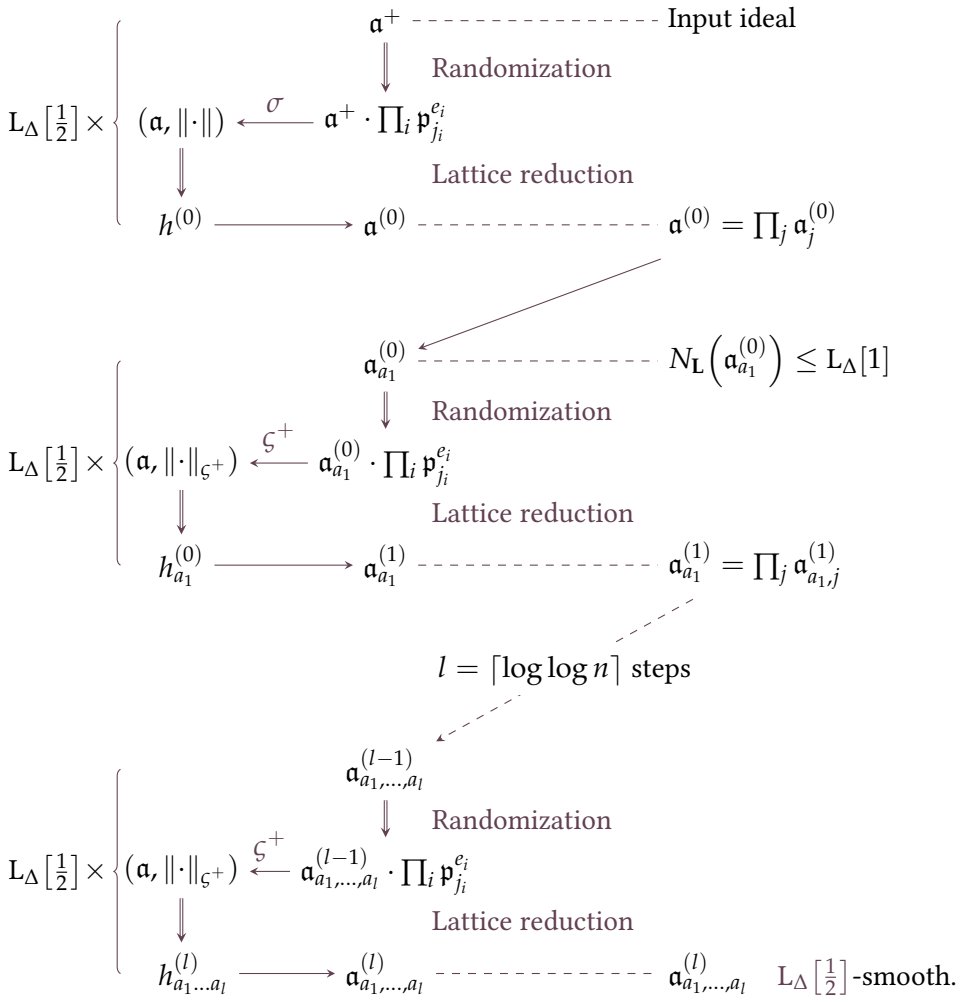


Figure 4: The descent algorithm.

6.2.6 Step 3: Case of $L_\Delta\left[\frac{1}{2}\right]$-smooth ideals

At this point, we have reduced the search for a generator of a principal ideal of large norm to the search for a generator of a principal ideal $\mathfrak{a}^s$ which is $L_\Delta\left[\frac{1}{2}\right]$-smooth. To tackle this final problem, we follow an approach similar to class group computation: take the set $\mathcal{B}$ of prime ideals of norm below $B > 0$ where $B = L_\Delta\left[\frac{1}{2}\right]$ and seek for relations of the form

$$v\mathscr{O}_{\mathbf{L}^+} = \prod_i \mathfrak{p}_i^{e_i}, \qquad \text{for } v \in \mathscr{O}_{\mathbf{L}^+},$$

for $\mathfrak{p}_i$ prime ideals of norm smaller than B. Formally this corresponds to looking for element in the kernel of the surjective map:

$$\begin{array}{ccccc} \mathbf{Z}^{|\mathcal{B}|} & \xrightarrow{\phi} & \mathcal{S}_{\mathcal{I}} & \xrightarrow{\pi} & \mathrm{Cl}(\mathscr{O}_{\mathbf{L}^+}) \\ (e_1, \cdots, e_{|\mathcal{B}|}) & \longmapsto & \prod_i \mathfrak{p}_i^{e_i} & \longmapsto & \prod_i [\mathfrak{p}_i]^{e_i}, \end{array}$$

Thanks to class group studies (see for instance [14, 28]), the relations heuristically form a full-rank sublattice of $\mathbf{Z}^{|\mathcal{B}|}$. Hence we need to find at least $\mathrm{L}_\Delta[1/2]$ linearly independent relations to generate this lattice.

The relation collection is performed in a similar way as in [13, 61]: due to the nice shape of the defining polynomial, the algebraic integers whose representation as polynomials in ζ have small coefficients also have small algebraic norms. Let us fix an integer $A > 1$. Then for any integers

$$(v_0, \ldots, v_{\frac{n}{2}-1}) \in \{-A, \ldots, A\}^{\frac{n}{2}},$$

we define the element

$$v = v_0 + \sum_{i \geq 1} v_i \left(\zeta^i + \zeta^{-i}\right)$$

The norm of this element in $\mathbf{L}^+$ is upper bounded by $\mathrm{L}_\Delta[1]$. Indeed, it corresponds to the square root of its norm in $\mathbf{L}$, which is $\mathrm{O}\left(\sqrt{n}^n\right) = \mathrm{L}_\Delta[1]$ by Hadamard's inequality. Then under Heuristic 6.1.1, the element v generates an ideal $v\mathscr{O}_{\mathbf{L}^+}$ that is $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$-smooth with probability $\mathrm{L}_\Delta\left[\frac{1}{2}\right]^{-1}$. This means that the expected number of independent algebraic integers to find one relation is $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$.

To bound the runtime of the algorithm, we also need to rely on another heuristic:

Heuristic 6.2.1. *There exists a value K that is negligible compared with $|\mathcal{B}|$ such that collecting $K \cdot |\mathcal{B}|$ relations suffices to obtain a relation matrix that has full-rank.*

It implies that there exists $K = \mathrm{o}\left(\mathrm{L}_\Delta\left[\frac{1}{2}\right]\right)$ such that collecting $K \cdot |\mathcal{B}|$ relations suffices to obtain a relation matrix that has full rank. We conclude that $\mathrm{L}_\Delta\left[\frac{1}{2}\right]^2 = \mathrm{L}_\Delta\left[\frac{1}{2}\right]$ independently drawn algebraic integers suffice to generate a full-rank matrix. Of course, the set of algebraic integers arising from the previous construction is large enough to allow such repeated sampling, because its size is $\mathrm{L}_\Delta[1]$. We store the relations in a $K|\mathcal{B}| \times |\mathcal{B}|$ matrix M, and store the corresponding algebraic integers in a vector v, as depicted as follows:

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{K|\mathcal{B}|} \end{pmatrix} \begin{matrix} \to \\ \to \\ \vdots \\ \to \end{matrix} \begin{pmatrix} M_{1,1} & \cdots & M_{1,|\mathcal{B}|} \\ M_{2,1} & \cdots & M_{2,|\mathcal{B}|} \\ \vdots & & \vdots \\ M_{K|\mathcal{B}|,1} & \cdots & M_{K|\mathcal{B}|,|\mathcal{B}|} \end{pmatrix} \Longrightarrow \forall i, v_i \mathscr{O}_{\mathbf{L}^+} = \prod_{j=1}^{|\mathcal{B}|} \mathfrak{p}_j^{M_{i,j}}.$$

The $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$-smooth ideal $\mathfrak{a}^s$ splits over the set $\mathcal{B}$, so that there exists a vector Y in $\mathbf{Z}^{|\mathcal{B}|}$ containing the exponents of the factorization

$$\mathfrak{a}^s = \prod_i \mathfrak{p}_i^{Y_i}.$$

As the relations stored in M generate the lattice of all elements of this form, the vector Y necessarily belongs to this lattice. Hence solving the equation $MX = Y$ yields a vector $X \in \mathbf{Z}^{K|\mathcal{B}|}$. From this vector, we can recover a generator of the ideal since:

$$\prod_i \mathfrak{p}_i^{Y_i} = \left(v_1^{X_1} \cdots v_{K|\mathcal{B}|}^{X_{K|\mathcal{B}|}}\right)\mathscr{O}_{\mathbf{L}^+}. \tag{6.3}$$

By construction, $N_{\mathbf{L}^+}\left(\mathfrak{a}^s\right) \leq \mathrm{L}_\Delta\left[\frac{\ell+1}{2}\right]$ so that the coefficients of Y are bounded by $\mathrm{L}_\Delta[0]$. Since solving such a linear system with Dixon's p-adic method [46] can be done in time $\mathrm{Poly}(d, \log\|M\|)$ where d is the dimension of the matrix and $\|M\| = \max|M_{i,j}|$ the maximum of its coefficients, we can find X in time $\mathrm{L}_\Delta\left[\frac{1}{2}\right]$.

6.3 ESTIMATION OF THE FULL COMPLEXITY

The overall runtime of our attack is $L_\Delta\left[\frac{1}{2}\right]$, that is a $2^{\mathrm{O}\left(\sqrt{n}\log n\right)}$ operations. We have already mentioned on-the-fly the complexity of most parts of our algorithm, but give in this section some details and provide a detailed analysis to estimate the second constant of the L notation.

6.3.1 Complements on the coarse complexity analysis

Concerning the reduction algorithms, using DBKZ on the first part of the HNF of the lattice, the block-size is always chosen as $\log L_\Delta\left[\frac{1}{2}\right]$ so that the complexity of the reduction procedure is in $L_\Delta\left[\frac{1}{2}\right]$. Our choice for the smoothness bound $B = L_\Delta\left[\frac{1}{2}\right]$ ensures that time $L_\Delta\left[\frac{1}{2}\right]$ suffices for the relation collection and linear system solution, as detailed in Section 6.2.6.

In addition, from the work of [58], the first part of the algorithm, corresponding to the reduction to the totally real subfield, is known to be polynomial time.

As a consequence the overall complexity is dominated by the descent step, which runs in time $L_\Delta\left[\frac{1}{2}\right]$. We now deepen the analysis in order to exhibit the second constant of this L notation.

6.3.2 Estimate of the second constant.

We recall that in our context, the extension degree n satisfies

$$n = \frac{\log\Delta}{\log\log\Delta}[1 + \mathrm{o}(1)].$$

6.3.2.1. *Bootstrapping phase.* Let us introduce a real constant $c_0 > 0$ which will be optimized to minimize the global cost of the algorithm. When using a block-size $\beta = c_0 \left[\frac{\log \Delta}{\log\log \Delta}\right]^{\frac{1}{2}}$ for the DBKZ reduction, the bootstrap of Algorithm 25 returns an ideal whose norm is bounded by $L_\Delta\left[\frac{3}{2}, \frac{1}{2c_0}\right]$ from which we can derive an ideal $\mathfrak{a}^{(0)}$ that is $L_\Delta\left[1, \left[\frac{1}{16c_0}\right]^{\frac{1}{3}}\right]$-smooth in time $L_\Delta\left[\frac{1}{2}, \left[\frac{9}{16c_0}\right]^{\frac{1}{3}}\right]$. This is a refinement of Theorem 6.2.2, according to the cost stated in Equation 6.1.

6.3.2.2. *Descent phase.* The analysis of the descent is done by refining the complexity of Theorem 6.2.3. At the k-th step of the descent, we have seen in Paragraph 6.2.5.2 that we take an ideal of norm $\mathrm{L}_\Delta\left[\frac{1}{2} + \frac{1}{2^{k+1}}\right]$. To precise this complexity, suppose that we are actually given as input an ideal of norm

$$L_\Delta\left[\frac{1}{2} + \frac{1}{2^{k+1}}, s_k\right],$$

for a certain constant s_k.

Then using the same algorithm an is Theorem 6.2.3, we are able to derive as output an ideal of norm

$$L_\Delta\left[\frac{1}{2} + \frac{1}{2^{k+2}}, s_{k+1}\right].$$

Each step has a runtime given by $L_\Delta\left[\frac{1}{2}, \frac{s_k}{2c_d s_{k+1}}\right]$, where $c_d > 0$ is a constant that depends on the dimension of the sublattice we are searching in. Finally, after $l = \lceil \log_2 \log n \rceil$ steps, we get an ideal that is $L_\Delta\left[\frac{1}{2} + \frac{1}{2^{l+1}}, s_l\right]$-smooth, and as $L_\Delta\left[\frac{1}{2} + \frac{1}{2^{l+1}}, s_l\right] = L_\Delta\left[\frac{1}{2}, e \cdot s_l\right]$, this suffices for our purposes.

6.3.2.3. *Linear algebra phase.* Let us fix $c_b = e \cdot s_l$ and $B = L_\Delta\left[\frac{1}{2}, c_b\right]$. We now have to handle of all the B-smooth ideals resulting from the descent. As a consequence of Landau's Prime Ideal Theorem [103], we know that the factor base composed of all prime ideals of norm below B has cardinality $B\log(B)^{-1}\big(1 + o(1)\big)$. The ideals generated by the algebraic integers v used in Step 3 have norm upper bounded by $L_\Delta\left[1, \frac{1}{4}\right]$ by construction. Hence we can find one that is B-smooth by testing on average $L_\Delta\left[\frac{1}{2}, \frac{1}{8c_b}\right]$ of them. As we want about $B\big(1 + o(1)\big)$ relations, the cost of this collection is given by $L_\Delta\left[\frac{1}{2}, c_b + \frac{1}{8c_b}\right]$. Eventually the remaining part is to solve the corresponding linear system. To do so, we can use for instance the Las-Vegas algorithm described by Storjohann in [155], whose complexity is in $L_\Delta\left[\frac{1}{2}, \omega \cdot c_b\right]$.

6.3.2.4. *Optimizing on c.* Combining the complexity of the relation collection and the linear algebra yields a final cost of

$$L_\Delta\left[\frac{1}{2}, \omega \cdot c_b\right] + L_\Delta\left[\frac{1}{2}, \frac{1}{8c_b}\right]$$

which is minimal when $c_b + \frac{1}{8c_b} = \omega \cdot c_b$, that is $c_b = \frac{1}{2\sqrt{2(\omega-1)}}$. Therefore the complexity of algorithm post-descent is in

$$L_\Delta\left[\frac{1}{2}, \frac{\omega}{2\sqrt{2(\omega-1)}}\right].$$

Now remark that we can set c_0 to satisfy: $\left[\frac{9}{16c_0}\right]^{\frac{1}{3}} = \frac{\omega}{2\sqrt{2(\omega-1)}}$ to obtain a bootstrap complexity also in

$$L_\Delta\left[\frac{1}{2}, \frac{\omega}{2\sqrt{2(\omega-1)}}\right],$$

yielding an ideal $\mathfrak{a}^{(0)}$ which is $L_\Delta\left[1, \frac{\omega}{2\sqrt{2(\omega-1)}}\right]$-smooth (that is, setting $s_0 = \frac{\omega}{2\sqrt{2(\omega-1)}}$).

For the subsequent steps, it is sufficient to fix

$$s_{k+1} = s_k \cdot \left(\frac{c_b}{e \cdot s_0}\right)^{\frac{1}{\ell}}$$

to reach $s_\ell = s_0 \cdot \frac{c_b}{e \cdot s_0} = c_b$. We can then fix c_d large enough to get a complexity of

$$L_\Delta\left[\frac{1}{2}, \frac{\omega}{2\sqrt{2(\omega-1)}}\right],$$

for the descent phase too.

Taking $\omega = \log_2 7$ (using for instance Strassen's fast multiplication algorithm) yields a runtime for our attack of $L_\Delta\left[\frac{1}{2}, 0.738\right] = 2^{1.066\sqrt{n}\log n}$.

Part III

CRYPTOGRAPHICAL PERSPECTIVES

In this final and shorter part we make use of the techniques introduced in Part II to tackle cryptanalytical problems. After getting through the context of lattice-based crytography, we expose the cryptanalysis of two schemes, where the reduction of algebraic lattices and the use of some algorithmic number theory is central.

7

A BIRD'S EYE VIEW ON LATTICE-BASED CRYPTOGRAPHY

This short historical chapter aims at providing a transition between the theoretical considerations of Part II and the applied research problems tackled in the final part of this manuscript.

We start by a brief panorama of the mutation encountered in cryptography in the second half of the XX$^{\text{th}}$ century, with focus on so-called "lattice-based cryptography". We try to enlighten how the algorithmic geometry of numbers has become an important tool for both cryptanalysts and cryptographers. Indeed since the seminal works of Ajtai [2], Euclidean lattices have become an interesting workplace for creating cryptographical schemes with rich features (such as *fully homomorphic encryption*, *graded encoding schemes*, *identity-based encryption*, ...). But lattices have also been a practical tool for cryptanalysis, not only for breaking the above mentioned lattice-based cryptography but also for classical systems as RSA.

7.1 BIRTH AND RISE OF ASYMMETRIC CRYPTOGRAPHY

7.1.1 From the secret-key paradigm...

Up to the second half of the XX$^{\text{th}}$ century, all cryptographic constructions were built upon the same paradigm, called today *symmetric cryptography*: we suppose that the persons who want to exchange secret data *share* the knowledge of a common secret. In a nutshell, using an encryption algorithm with the secret key, data is converted to a form that is unintelligible by anyone who does not possess this secret key to decrypt it. Once the intended recipient, who possesses the key, has the message, the algorithm reverses its action so that the message is returned to its original and understandable form. It is still a very vivid technological area as having incomparable practical performances.

7.1.2 ... to the public-key paradigm

However, the caveat of this methodology lies in the absolute necessity for the two parties to *agree* beforehand on the shared secret. In 1976, Diffie and Hellman in the foundational[1] paper *New directions in cryptography* [44]

1 It was however revealed in 1997 that James H. Ellis, Clifford Cocks, and Malcolm J. Williamson of GCHQ, the British intelligence agency, had previously shown how public-key cryptography could be achieved in 1969.

demonstrated an exchange method allowing two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. Such procedure uses a pair of keys instead of a unique shared key: public key which may be sent over any public channel, and private key which is known only by the owner. By contrast with the symmetric cryptography, in a public key system, the public keys can be disseminated widely and openly—and only the private key needs to be kept secure by its owner.

Two of the best-known uses of public key cryptography are the encryption schemes and the signature schemes. Since we are going to provide a cryptanalysis of an encryption scheme [153] and of a signature [47], we provide a generic definition of such primitives for completeness purposes.

7.1.2.1. *Public key encryption.* A message is encrypted with a recipient's public key. The message cannot be decrypted by anyone who does not possess the matching private key, who is thus presumed to be the owner of that key and the person associated with the public key. This aims at ensuring the confidentiality of the data. Such a scheme consists of three different algorithms:

Key generation: selects a private key (usually uniformly at random) from the set of secret keys. The algorithm outputs it as well as the corresponding public key.

Encryption algorithm: produces a ciphertext when fed with a message and a public key.

Decryption: Given the ciphertext and secret key, retrieves the original message.

7.1.2.2. *Signatures schemes.* A message is signed with the sender's private key and can be verified by anyone who has access to the sender's public key. This verification proves that the sender had access to the private key, and therefore is likely to be the person associated with the public key. This also ensures that the message has not been tampered with, as a signature is mathematically bound to the message it originally was made with, and that verification will fail for any other message. A signature scheme consists of three different algorithms:

Key generation: selects a private key (usually uniformly at random) from the set of secret keys. The algorithm outputs it as well as the corresponding public key.

Signing algorithm: produces a signature when fed with a message and a private key.

Verifcation: Given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

7.1.3 One way functions and hard problems

The public key paradigm essentially relies on the existence of *one-way functions*, that is a function F acting on binary strings such that there is a polynomial-time algorithm which maps any string $r \in \{0,1\}^*$ to $F(r)$ and such that for every probabilistic polynomial-time algorithm A, constant $c > 0$, and sufficiently large $n \in \mathbf{N}$, we have:

$$\Pr_{v=F(r);r\sim\mathcal{U}(\{0,1\}^n)}\left[F(A(v)) = v\right] < n^{-c}.$$

This means that such F is easy—by the polynomial time mapping to the image of F—to compute on every input, but hard to invert given the image of a random input. While the definition of one-way functions does not involve any secret key, it was shown in a large amount of research, mainly through the connection to pseudorandomness enabled by the Goldreich–Levin theorem [68], that the existence of one way functions is equivalent to the existence of many cryptographic primitives including pseudorandom generators, digital signatures, commitment schemes and zero knowledge proofs for every language in NP. Remark that for any one-way function F, the inversion of F is *hard* to compute, by definition, but is actually easy to check, by just computing F on it. Thus, the existence of a one-way function implies that that P $\neq$ NP. However, it is not known whether the fact that P $\neq$ NP actually implies the existence of one-way functions. Indeed, we do not know if such functions exist: several candidates have been proposed and are supposed to be one-way, but extensive research has so far failed to produce an efficient inverting algorithm for any of them.

7.2 NEW TOOLS FOR NEW CONSTRUCTIONS

This conceptual change also required the development of new mathematical tools to make such exchanges possible, that is by giving candidates of one-way functions. Symmetric cryptography was very close by design to the conception of hash functions, with iterations of fast non-linear Boolean functions, whereas the asymmetric cryptography requires tools from number theory, from the elementary theory of cyclic rings $\mathbf{Z}/(n)$ (such as RSA [141]), elliptic curves [97], coding theory [118] or *lattices*.

7.3 ON LATTICE BASED CRYPTOGRAPHY

Although the introduction of number theory for cryptographical constructions goes back to the early 70s, the use of lattices only appears two decades later with the work of Ajtai in 1996 [2], who constructed a hash function whose security relies on the *short integer solution* (SIS) problem. In this paper he also demonstrated that the average-case of the SIS problem was at least as hard as the worst case of the SVP problem.

A milestone in the history of lattice-based public-key cryptography came from the work of Regev in [139], who introduced a new hard problem: the

Learning with Errors problem (LWE), inspired by the classical learning problems on vector spaces. This work showcases an encryption scheme whose security is proved under the worst-case hardness assumptions of the SIVP problem.

In order to optimize the lattice-based constructions, it has been proposed to migrate from Euclidean lattices to more structured ones, such as ideal lattices [114], and more generally algebraic lattices [105]. Indeed, the algebraic structure of these lattices allows to handle their elements in very compact forms such as vectors of integral polynomials. For instance an ideal lattice can be represented by a couple of polynomials instead of a full rank integral matrix.

Further, many lines of work have been initiated to optimize the efficiency of lattice scheme as well as enhancing the cryptographic possibilities of the scheme. A breakthrough instance of this direction of research is the creation of the first *fully homomorphic encryption* (FHE) scheme by Craig Gentry in 2009 [62]. FHE allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext.

7.4 ALGORITHMIC GEOMETRY OF NUMBERS AS A CRYPTANALYTICAL TOOLKIT

7.4.1 Theoretical security arguments from lattices

If the use of lattices as a constructive tool for the cryptographer is a somewhat recent matter, it is not the case in cryptanalysis. Of course lattice-based schemes are attacked through lattice reduction algorithms, as their security is proved by reduction to SVP, CVP or SIS problems, which are solved in practice by performing lattice reduction. But the range of applications of lattice techniques is wider than that.

Indeed, as reduction techniques allow finding short or small modular linear dependencies among a set of vectors, a natural use of lattice reduction is the cryptanalysis of so-called *knapsack problems*, specific setting. A more number-theoretical example is the cryptanalysis of RSA when part of the secret key is known by the use of the so-called *Coppersmith technique* [38]. It consists in a method to find small integer zeroes of univariate or bivariate polynomials modulo a given integer. It uses reduction algorithms to find a polynomial over $\mathbf{Z}$ that has the same zeroes as the target polynomial but with smaller coefficients.

In all the cases where lattices are used for cryptanalysis, the goal remains the same: find a short enough vector in an ad hoc lattice, which will retrieve the secret key, or at least will allow the construction of a trapdoor to break the attacked cryptosystem. Therefore, enhancing lattice reduction algorithms impacts directly the cryptanalysis of numerous schemes.

As mentioned, modern lattice-based schemes are built on algebraic lattices, such as the Bliss or Dilithium signatures [50, 115], for instance. Hence,

the design of fast lattice reduction for algebraic lattices is a natural matter for the security evaluation of lattice-based cryptography.

7.4.2 On the practical security of real-world cryptography

As hinted above, the theoretical security evaluation of a public key primitive gets through the theoretical analysis of a presumably hard problem. However, in the context of real-world implementation of a scheme, one can exploit more than its abstract specification, by exploiting the whole physical fingerprint the algorithm lefts in the physical world, during its computation. A *side-channel attack* is any kind of attack using pieces of information gained from the implementation of a physical system, rather than weaknesses of the algorithm itself. Such leaks can be found for instance in the cache[2], read as timing information[3], power consumption[4], electromagnetic leaks or even sounds. Hence, providing a theoretically secure algorithm is not sufficient to ensure a secure implementation of the scheme. However, the exploitation of side-channel traces is usually not direct and often requires non trivial techniques to retrieve secret elements, as we demonstrate in the next chapter.

2 These attacks are based on attacker's ability to monitor cache accesses made by the victim in a shared physical system as in virtualized environment.

3 These attacks are mounted by measuring how much time various computations such as, say, comparing an attacker's given password with the victim's unknown one take to perform.

4 These attacks make use of varying power consumption by the hardware during computation.

8

CRYPTOGRAPHICAL ATTACKS BY NUMBER THEORY

This final chapter showcases how the algorithm exposed in Part II can be applied to perform practical cryptanalysis of lattice-based schemes, namely *fully-homomorphic encryption* and *signatures*. We start by looking at the application of the PIP algorithm of Chapter 6 solving algorithm to perform a full key recovery of the fully-homomorphic encryption scheme of Smart and Vercauteren. Our final landmark is an attack in the context of *side-channel attacks* on the BLISS signature scheme, which crucially relies on fine grained-ideal manipulation, integer factorization and on the Gentry-Szydlo algorithm presented at the end of Chapter 5.

8.1 A KEY RECOVERY ON SMART AND VERCAUTEREN'S FHE SCHEME

8.1.1 Relation the principal ideal problem

Among all the Fully Homomorphic Encryption (FHE) schemes proposed in the last decade, the security of a couple of them directly collapses if it is possible to find relatively short generators in principal ideals. This is the case of the proposal of Smart and Vercauteren [153], which is a simplified version of the original scheme of Gentry [62]. Other schemes based on the same security assumption include the Soliloquy scheme of Campbell, Groves, and Shepherd [29] and candidates for multilinear maps [58, 106]. More formally, the underlying—presumably hard—problem is derived from the SVPintroduced in Chapter 6: it is the SG-PIP (Short Generator-Principal Ideal Problem): given some $\mathbf{Z}$-basis of a principal ideal with a promise that it possesses a "short" generator g for the Euclidean norm, find this generator or at least a short enough generator of this ideal. The strategy to address this problem roughly splits into two main steps:

1. Principal ideal problem: given the $\mathbf{Z}$-basis of the ideal, find a generator, not necessarily short, that is $g' = g \cdot u$ for a unit u, that is solve an instance of PIP.

2. Reduction from g', find a short generator of the ideal.

Several results have allowed to deal with the second step. Indeed, Campbell, Groves, and Shepherd [29] claimed in 2014 an efficient—although unproven—solution for power-of-two cyclotomic fields, confirmed by experiments conducted by Schanck [144] in 2015. Eventually, the proof was provided by Cramer, Ducas, Peikert, and Regev [42], together with an extension to all

prime-power cyclotomic fields. We proposed an extension to this algorithm as a byproduct of the design of the algorithms of Chapter 5 for *arbitrary* cyclotomic fields and decreased the complexity to quasilinear.

As a direct illustration of the algorithm presented in Chapter 6, we showcase an attack on the scheme that Smart and Vercauteren describe in [153], which leads to a *full key recovery*. We recall in Algorithm 26 the key generation process in the case of power-of-two cyclotomic fields. This instantiation is the one chosen by the authors for presenting their implementation results.

Algorithm 26 — Key-Generation

Input : The security parameter $n = 2^m$.
Output : A pair $(\mathrm{sk}, \mathrm{pk})$ of secret/public keys.

$\Phi_{2n}(X) \leftarrow X^n + 1$ as the polynomial defining the cyclotomic field $\mathbf{L} = \mathbf{Q}(\zeta_{2n})$
do
  $G(X) \leftarrow 1 + 2 \cdot S(X)$ for $S(X)$ of degree $n-1$ with coefficients absolutely bounded by $2^{\sqrt{n}}$
  $g \leftarrow G(\zeta_{2n}) \in \mathscr{O}_{\mathbf{L}}$
while $N_{\mathbf{L}/\mathbf{Q}}\big(G(\zeta_{2n})\mathscr{O}_{\mathbf{L}}\big)$ *is prime*
return $\big(\mathrm{sk} = g, \mathrm{pk} = \mathrm{HNF}\big(g\mathscr{O}_{\mathbf{L}}\big)\big)$

Remark. *The public key can be any* $\mathbf{Z}$*-basis of the ideal generated by* g*, or even a two-elements representation of this ideal. Indeed, the original paper of Smart and Vercauteren [153] provides the public key as a pair of elements that generates the lattice, for compactness purposes.*

As our attack consists in a full secret-key recovery, only based on the public key, the encryption and decryption procedures are irrelevant for our purposes. As such we omit their presentation in this present chapter. Even though this work is more concerned with the principal ideal problem rather than the reduction step, we emphasize the fact that the short generator resulting from this reduction is the key modulo the multiplicative action of the roots of unity of the number field. This is not an issue, since all these keys are equivalent with regard to the decryption procedure. In addition, in this precise construction of the Smart and Vercauteren FHE scheme, the only odd coefficient of $G(X)$ is the constant one, so that we may recover the exact generator g readily—as multiplying by a primitive root of unity acts as a simple coefficient shift.

8.1.2 Implementation results

In [153, Section 7], security estimates of Smart and Vercauteren's scheme are given for parameters $n = 2^m$ for $8 \leq m \leq 11$, since the authors were unable to generate keys for larger parameters. Our implementation allows us to find a secret key from the public key for $n = 2^8 = 256$ in less than a day: the

code runs with Pari/GP [132], with an external call to fplll [159], and all the computations are performed on an Intel(R) Xeon(R) CPU E3-1275 v3 @ 3.50GHz with 32GB of memory. The large storage requirements are due to the Gentry-Szydlo algorithm.

We perform the key generation as in Algorithm 26. We then obtain a generator for the ideal as a polynomial in $\zeta = \zeta_{512}$, of degree 255 and coefficients absolutely bounded by $2^{\sqrt{256}} + 1 = 65537$. That corresponds to ideals whose norms have about 4800 bits on average, which is below the bound 6145 obtained from the following claim:

Lemma 8.1.1. *Let* $\mathbf{L} \cong \mathbf{Q}[X]/(T)$ *be a number field, defined by a unitary irreducible polynomial* T*, and* θ *be the image of* X *in* $\mathbf{L}$*. For an algebraic integer* $x = P_x(\theta)$ *for* $P_x \in \mathbf{Z}[X]$*, then*

$$|N_{\mathbf{L}/\mathbf{Q}}(x)| = Res(T, P_x) \leq (n+1)^{m/2}(m+1)^{n/2} H(P_x)^n H(T)^m,$$

where $n = [\mathbf{L} : \mathbf{Q}]$ *and* $m = \deg P_x = \deg x$*.*

Proof. By definition of the resultant, since T is unitary, we have

$$\mathrm{Res}(T, P_x) = \prod_{\rho \in \mathrm{Root}(T)} P_x(\rho).$$

Remark that the $P_x(\rho)$ are the conjugates of $P_x(\theta)$ in the algebraic closure of $\mathbf{L}$, so that $N(x) = \prod_{\rho \in \mathrm{Root}(T)} P_x(\rho)$. The upper bound is a direct consequence of Hadamard's bound on the Sylvester matrix defining $\mathrm{Res}(T, P_x)$ as a determinant. ∎

This bound is nonetheless above the size given in [153] (4096). As for all timings in this section, we have derived a set of 10 keys, and the given time is the average one. Thus, deriving a secret key takes on average 30 seconds. We test 1381 algebraic integers resulting in ten that have a prime norm. Then the public key is derived from the secret key in about 96 seconds.

While, in theory, the first reduction to the totally real subfield seems to be of limited interest, it is clearly the main part of the practical results: indeed, it reduces in our example the size of the matrices involved from 256×256 to 128×128. As we know that the quality of lattice-reduction is getting worse as the dimension grows, this part is the key point of the algorithm. The attack essentially corresponds to the Gentry-Szydlo algorithm described in Section 5.6.1 together with the trick explained in Paragraph 6.2.3.1, in order to output the element u and a basis of the ideal $\mathfrak{a}^+$ generated by $g + \bar{g}$. This part of the algorithm runs in less than an hour, and requires about 24GB of memory.

At this point, we put aside u and only consider the ideal $\mathfrak{a}^+$. Our goal is to recover one generator of this ideal, after which a multiplication by $\frac{1}{1+u}$ leads to a generator of the input ideal. The method we have presented is to reduce step by step the norms of the ideals involved by performing lattice reductions. However, we observe that for the cases we run, the first reduction suffices: the short vector we find corresponds to a generator. We make use

of the BKZ algorithm implemented in fplll [159], with block-size 30, requiring between 2 and 4 hours of computation. This surprisingly good practical behavior of the lattice reduction is a consequence of the conjunction of two facts. One is the dimension of the lattices involved—medium dimensions allow better practical output bounds than the theoretical worst case—and the other one are the favorable properties of the geometry of the considered ideals.

In addition to the good behavior of this reduction, the generator we found is already small, by construction. More precisely, it corresponds to $g + \bar{g}$, up to a factor that is a power of ζ. Hence, we recover $g \cdot \zeta^i$ thanks to u and the decoding algorithm presented in Section 5.1 turns out to be unnecessary for our application. The key recovery is already completed after these first two steps. Nevertheless, we implemented this part along with a method to find the actual private key (up to sign). Indeed, because all its coefficients are even except the constant one, it is easy to identify the power of ζ that appears as a factor during the computation.

All in all, given the public key $\mathfrak{a}$, the *practical* attack reduces to the following steps:

1) Reduction to the maximal totally real subfield: Reduce PIP for $\mathfrak{a}$ to the PIP for an ideal $\mathfrak{a}^+$ in the maximal totally real subfield, with the technique of Section 6.2.3 coupled with the Gentry-Szydlo algorithm of Section 5.6.1.

2) Lattice reduction: Find a short element $\alpha \in \mathfrak{a}$ such that

$$\mathfrak{a}^+ = \alpha\mathfrak{a}',$$

and $\mathfrak{a}'$ is a smooth ideal (precisely, $L_\Delta\left[\frac{1}{2}\right]$-smooth, as in Chapter 6). This element is discovered by a DBKZ-reduction on the ideal lattice corresponding to $\mathfrak{a}$.

3) Collection relation and resolution on $\mathfrak{a}$: Using the relation collection of Section 6.2.6 find a generator β of $\mathfrak{a}'$ by linear algebra.

4) Key recovery: Shift the coefficients of the $\alpha\beta$ to recover the private key.

To conclude, for the parameter $n = 2^8$, the time of the key recovery is below less than 5 hours, and the main part of the computation comes from the lattice reduction part.

Remark (On the theoretical complexity of the attack.)**.** *The PIP-solving algorithm of Chapter 6 has a complexity in $L_\Delta\left[\frac{1}{2}\right]$ in the discriminant that represents the size of the number field involved. However, it is important to ascertain that the parameters of the keys have $n^{\frac{3}{2}}$ bits. Therefore we present an algorithm that is "sort of" $L\left[\frac{1}{3}\right]$ in the size of the inputs.*

8.2 A SIDE-CHANNEL ATTACK ON BLISS SIGNATURE SCHEME

In the early days of lattice-based cryptography, several signature schemes with heuristic security were proposed, most notably GGH and NTRUSign (see [67] and [83]), but despite several attempts to patch them, they turned out to be insecure: it was found that the distribution of generated signatures leaks statistical information about the secret key, which can be exploited to break these schemes and their variants [65, 66, 129]. The most common approach to obtain efficient, provably secure lattice-based signatures in the random oracle model is the "Fiat–Shamir with aborts" paradigm introduced by Lyubashevsky [112] (it coexists with the GPV hash-and-sign paradigm relying on lattice trapdoors [63], which has some theoretical benefits compared to Fiat–Shamir, but tends to result in less efficient implementations [48]). Lyubashevsky's approach is an extension of the usual Fiat–Shamir transformation which uses *rejection sampling* to make sure that generated signatures have a distribution independent of the secret key, and avoid the statistical pitfalls of schemes like NTRUSign. More precisely, the underlying identification protocol achieves its honest-verifier zero-knowledge property by aborting some of the time, and signatures are produced by re-running that protocol with random challenges until it succeeds.

Several instantiations of this paradigm have been proposed [3, 70, 84, 113], targeting various output distributions for signatures, but one of the most popular among them is certainly the BLISS signature scheme proposed by Ducas et al. [49]. It is possibly the most efficient lattice-based signature scheme so far, boasting performance comparable to common implementations of RSA and ECC-based signatures, such as the one in OpenSSL. Signature and public-key size are a few times larger than RSA (and about one order of magnitude bigger than ECC); signature generation is comparable to ECC and beats RSA by an order of magnitude; and signature verification is similar to RSA and faster than ECC by an order of magnitude.

This efficiency is achieved in particular through the use of Gaussian noise, and a target distribution for signature that has a *bimodal Gaussian* shape. This makes the rejection sampling step for BLISS somewhat tricky to implement, particularly on platforms where evaluating transcendental functions to a high precision is impractical. However, the authors of [49] proposed an efficient technique to carry out this rejection sampling based on iterated Bernoulli trials. This technique is used, in particular, in the embedded implementations of BLISS described in [85, 137].

8.2.1 Description BLISS signature

One can give a simplified description of the scheme as follows. We are working inside the ring of integer $\mathcal{O}_{\mathbf{L}}$ of a cyclotomic field $\mathbf{L}$ whose conductor is a power-of-two. The public key is an NTRU-like ratio of the form $a_q = s_2/s_1 \bmod q$, where the signing key polynomials $s_1, s_2 \in \mathcal{O}_{\mathbf{L}}$ are small and sparse. To sign a message μ, one first generates commitment values

$y_1, y_2 \in \mathcal{O}_{\mathbf{L}}$ with normally distributed coefficients, and then computes a hash c of the message μ together with $u = -a_q y_1 + y_2 \bmod q$. The signature is then the triple (c, z_1, z_2), with $z_i = y_i + s_i c$. The rejection sampling to ensure that the distribution of z_i is independent of the secret key. Verification is possible because $u = -a_q z_1 + z_2 \bmod q$.

The real BLISS scheme, described in Algorithm 27, includes several optimizations on top of the above description. In particular, to improve the repetition rate, it targets a bimodal Gaussian distribution for the z_i's, so there is a random sign flip in their definition. In addition, to reduce key size, the signature element z_2 is actually transmitted in compressed form $z_2^\dagger$, and accordingly the hash input includes only a compressed version of u. The random oracle H takes its values in the set of polynomials in $\mathcal{O}_{\mathbf{L}}$ with 0/1 coefficients and Hamming weight exactly κ, for some small constant κ. We refer to the original paper [49] for the definition of notation like M, ζ, N_κ and $\lfloor \cdot \rceil_d$, as they are not relevant for our purposes.

Algorithm 27 — BLISS signature algorithm

Input : μ, $pk = a_1$, $sk = S$

1 $y_1 \leftarrow \mathcal{D}^n$, $y_2 \leftarrow \mathcal{D}^n$
2 $u = \zeta \cdot a_1 \cdot y_1 + y_2 \bmod 2q$
3 $c \leftarrow H(\lfloor u \rceil_d \bmod p, \mu)$
4 Choose a random bit b
5 $z_1 \leftarrow y_1 + (-1)^b s_1 c$
6 $z_2 \leftarrow y_2 + (-1)^b s_2 c$
7 **restart** to step 2 except with probability
$1/\left(M \exp(-\|Sc\|^2/(2\sigma^2)) \cosh(\langle z, Sc\rangle/\sigma^2)\right)$
8 $z_2^\dagger \leftarrow (\lfloor u \rceil_d - \lfloor u - z_2 \rceil_d) \bmod p$
9 **return** $(z_1, z_2^\dagger, c)$

8.2.1.1. *Implementation of the BLISS rejection sampling.* It is essential for the security of the scheme that the distribution of signatures is essentially statistically independent of the secret signing key. This is achieved using the rejection sampling step 6 of algorithm SIGN, as described in Algorithm 27.

To implement this rejection sampling in practice, one needs to be able to efficiently sample from Bernoulli distributions of the form $\mathcal{B}_{\exp(-x/f)}$ and $\mathcal{B}_{1/\cosh(x/f)}$ for some fixed constant f and variable integers x (where $\mathcal{B}_p$ denotes the Bernoulli distribution of parameter p, which outputs 1 with probability p and 0 otherwise).

Algorithm 28 — BLISS rejection sampling

$x \leftarrow K - \|Sc\|^2$
Sample $a \leftarrow$ **SampleBernExp**(x)
if $a = 0$ **then Restart** Algorithm 27
$x \leftarrow 2 \cdot \langle z, Sc \rangle$
if $x < 0$ **then** $x \leftarrow -x$
Sample $a \leftarrow$ **SampleBernCosh**(x)
if $a = 0$ **then Restart** Algorithm 27

This can in principle be done by computing the rejection probability every time with sufficient precision and by comparing it to uniformly sampled randomness in a suitable interval, but such an approach is quite costly, especially on constrained devices, as it relies on the evaluation of transcendental functions to arbitrary precision. Therefore, BLISS relies on an alternate approach, which is described in [49, §6] and can be implemented based on sampling Bernoulli distributions $\mathcal{B}_{c_i}$ for a few precomputed constants c_i.

The idea is as follows. To sample from $\mathcal{B}_{\exp(-x/f)}$, one can consider the binary expansion $\sum x_i \cdot 2^i$ of x, and let $c_i = \exp(-2^i/f)$. Then one has $\exp(-x/f) = \prod_{x_i=1} c_i$. As a result, sampling from $\mathcal{B}_{\exp(-x/f)}$ can be done by sampling from each of the $\mathcal{B}_{c_i}$; if all the resulting samples are 1, return 1, and 0 otherwise. This can even be done in a lazy manner, as described in algorithm **SampleBernExp** in Algorithm 29.

Algorithm 29 — Samplers

Function *SampleBernExp*:
  Input : $x \in [0, 2^\ell) \cap \mathbf{Z}$
  Output : Sample from $\mathcal{B}_{\exp(-x/f)}$, $x \in [0, 2^\ell)$.
  for $i = 0$ *to* $\ell - 1$ **do**
    if $x_i = 1$ **then** Sample $a \leftarrow \mathcal{B}_{c_i}$
    if $a = 0$ **then return** 0
  end for

Function *SampleBernCosh*:
  Input : $x \in [0, 2^\ell) \cap \mathbf{Z}$
  Output : Sample from $\mathcal{B}_{\cosh(-x/f)}$, $x \in [0, 2^\ell)$.
  Sample $a \leftarrow \mathcal{B}_{\exp(-x/f)}$
  if $a = 1$ **then return** 1
  Sample $b \leftarrow \mathcal{B}_{1/2}$
  if $b = 1$ **then restart**
  Sample $c \leftarrow \mathcal{B}_{\exp(-x/f)}$
  if $c = 1$ **then restart**
  return 0

In addition, one can show that sampling from $\mathscr{B}_{1/\cosh(x/f)}$ can be done by repeated sampling from $\mathscr{B}_{\exp(-x/f)}$ and $\mathscr{B}_{1/2}$, as described in algorithm **SampleBernCosh** in Algorithm 29 (the correctness of that method is proved as [49, Lemma 6.3]). The algorithm has an a priori unbounded number of iterations, but the expected number of calls to $\mathscr{B}_{\exp(-x/f)}$ is less than 3.

Concretely, the BLISS rejection sampling is thus implemented as follows. The denominator f in **SampleBernExp** and **SampleBernCosh** is set to $2\sigma^2$, and the scaling factor M for the rejection sampling is taken of the form $\exp(K/f)$ for some integer K. Then, step 6 of **Sign** in Algorithm 27 actually consists of the instructions described in Algorithm 28.

8.2.2 Exploiting the norm leakage

Let us suppose we can have access by single power analysis[1] (SPA) to the bits of $\|Sc\|^2$ in the final computation of the rejection sampling. Recalling that $Sc = (s_1c, s_2c)^T$, we have $\|Sc\|^2 = \langle s_1c, s_1c\rangle + \langle s_2c, s_2c\rangle$ and thus this norm can be seen as $C^T \cdot \Sigma^T \cdot \Sigma \cdot C$, where $C = (c, c)^T$ and

$$\Sigma = \left[\begin{array}{c|c} S_1 & 0 \\ \hline 0 & S_2 \end{array}\right],$$

for c being the vector encoding of the polynomial c, and S_1 (resp. S_2) being the skew-circulant matrix encoding the polynomial s_1 (resp. s_2). Let X be the matrix $\Sigma^T \cdot \Sigma$. Then, recovering the value $\|Sc\|^2$ yields an equation of the shape:

$$c^T \cdot X \cdot c = \|Sc\|^2. \tag{8.1}$$

This equation can be viewed as a row of a linear system whose unknowns are the coefficients of the secret-dependent matrix X. Remark that X is a block matrix of shape $\mathrm{Diag}\left(X^{(1)}, X^{(2)}\right)$ where $X^{(i)}$ are circulant matrices of first line

$$\left(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \cdots, x_{m/2+1}^{(i)}, 0, -x_{m/2+1}^{(i)}, \cdots, -x_3^{(i)}, -x_2^{(i)}\right)$$

as product of two conjugate skew-circulant matrices. Thus, only $2 \times m/2 = m$ distinct unknowns are actually present in X. As a consequence, we only need m linearly independent equations to fully recover the matrix X. Once recovered, we therefore get access to the submatrices $S_1^T \cdot S_1$ and $S_2^T \cdot S_2$. By definition, these matrices corresponds to the encoding of the polynomial $s_1 \cdot \bar{s}_1$ and $s_2 \cdot \bar{s}_2$, that is the relative norm of the secrets s_1 and s_2 in the maximal real subfield. This is an instance of so-called *relative norm equation*. In an arbitrary number field, the problem of recovering an element from its norm over a subfield is a hard problem of number theory and its complete solution requires an oracle to the PIP, yielding subexponentials algorithms [152]. However in the cyclotomic fields we are considering, we can avoid the use

1 SPA involves visual examination of graphs of the current used by a device over time to notice the spikes of consumption, on a digital oscilloscope for instance.

of this oracle. Precisely, the resolution is performed using a generalization of the Howgrave-Graham–Szydlo algorithm [87], detailed in Section 8.2.3. Suppose that one value $s_1 \cdot u_1$ or $s_2 \cdot u_2$ (with u_1, u_2 roots of unity), we can use our knowledge of the public key $s_2/s_1 \bmod q$ to recover candidates for the other part of the secret (once again up to unity). Hence, when candidate secrets are determined, we can discriminate valid keys among them by checking their sparsity and polynomial height, to satisfy the conditions imposed by the key generation procedure. The whole attack is described in Algorithm 30.

Algorithm 30 — Algebraic side-channel attack

1. Collect traces $(c^{(k)}, \|Sc^{(k)}\|^2)_k$ until the matrix $\mathcal{C}$ corresponding of the corresponding system is full-rank.
2. Solve the linear system $\mathcal{C} \cdot X = (\|Sc^{(1)}\|^2, \ldots, \|Sc^{(k)}\|^2)^T$.
3. Call Algorithm 31 on either $s_1 \cdot \overline{s_1}$ or $s_2 \cdot \overline{s_2}$ to recover s_1 and s_2 up to a root of unity.

The mostly costly part of this attack is the resolution of the norm equation, which in particular requires a norm factorization over the integers. In order to estimate the cost of this factorization step, one can bound the algebraic norm of the secret element. Let s be one of the secret elements s_1, s_2. Note that $N_{\mathbf{L}/\mathbf{Q}}(s)$ is equal to the resultant of s (explicitly, the lift of s in $\mathbf{Z}[X]$ from its representation in $\mathscr{O}_{\mathbf{L}}$). From Lemma 8.1.1 we have: $|N_{\mathbf{L}/\mathbf{Q}}(s)| \leq 2^{\frac{n}{2}} \left(\sqrt{\delta_1^2 + 4\delta_2^2 n}\right)^n$, yielding directly that

$$\log |N_{\mathbf{L}/\mathbf{Q}}(f)| \leq \frac{n}{2}\left(\log\left(n\sqrt{\delta_1^2 + 4\delta_2^2}\right) + 1\right).$$

Table 8.1 compiles the theoretical bound and the average practical results for the various proposed security parameters.

We can see that these integers are typically too large to be factored in practice. Since the success of the attack depends on the ability to factor the norm, we are only able to attack a fraction of the whole space of private keys, for which the factorization is easy. A particular class of them is the set of keys whose norm is a *B-semi-smooth* integer, that is a composite number $p \cdot b$, where p is prime and b is B-smooth for a non-negative integer B. As already remarked, the recovery of either s_1 or s_2 is sufficient to recover the full secret. Hence, the above-described attack becomes tractable as soon as one of the norm $N_{\mathbf{L}/\mathbf{Q}}(s_1), N_{\mathbf{L}/\mathbf{Q}}(s_2)$ is semi-smooth. This means that the probability of getting a weak key is twice the probability of one of the constituting part of the private key to have a semi-smooth norm. Practical estimations of the fraction of keys with semi-smooth norms are presented in Table 8.2.

Note that the entire attack is actually known to run on average in polynomial time, except, classically, the factorization of the norm.

Table 8.1: Estimation of the absolute norms of BLISS secret keys for the security parameters of [49] (experimental averages over 2000 keys per set).

			Bitsize of $N_{\mathbf{L}/\mathbf{Q}}(f)$	
	n	(δ_1, δ_2)	theoretical	exp. avg.
BLISS-0	256	$(0.55, 0.15)$	1178	954
BLISS-I	512	$(0.3, 0)$	2115	1647
BLISS-II	512	$(0.3, 0)$	2115	1647
BLISS-III	512	$(0.42, 0.03)$	2332	1866
BLISS-IV	512	$(0.45, 0.06)$	2422	1957

Table 8.2: Estimation of the proportion weak BLISS secret keys, namely those for which at least one of s_1 or s_2 has B-semi-smooth norm, for the security parameters of [49] and several choices of B. Estimates obtained by sampling 2000 secret keys per parameter set and testing their norm for semi-smoothness by trial division.

	n	$B = 2$	$B = 5$	$B = 65537$	$B = 655373$	$B = 6553733$
BLISS-0	256	4%	6%	7.6%	12%	13%
BLISS-I/II	512	2%	3%	4%	5.6%	7.4%
BLISS-III/IV	512	1.5%	2%	3.5%	4%	5%

Remark (An interesting feature)**.** *Amusingly, this means that the attack becomes* quantumly *fully polynomial: this is an interesting feature for an attack targeting a* post-quantum *scheme!*

The entire attack was implemented in PARI/GP, including the generalized Howgrave-Graham–Szydlo algorithm and the Gentry–Szydlo algorithm. To the best of our knowledge, this was the first full implementation of this algorithm. It allows to tackle the problem of solving norm equations in dimension up to[2] 512. Experiments were conducted with this implementation to obtain the running time of the attack and presented in Table 8.3, on a quad-core Intel I7-8650 CPU.

2 In their original paper, Howgrave-Graham and Szydlo were limited to smaller dimension (up to 100) and did not implement all the possible cases occurring in the algorithm.

Table 8.3: Average running time of the attack for various field sizes n. The BLISS parameters correspond to $n = 256$ and $n = 512$.

Field size n	32	64	128	256	512
CPU time	0.3 s	1 s	5 min.	3h20min.	1 Day

8.2.3 Howgrave-Graham–Szydlo Algorithm in Power-of-Two Cyclotomic Fields

We now present a generalization of the Howgrave-Graham-Szydlo algorithm to *power-of-two cyclotomic fields* since the BLISS signatures works over this class of fields. The original procedure solves the problem of recovering an element f of the ring of integers of a cyclotomic field of prime conductor l given its relative norm $f \cdot \overline{f}$ factorization. This problem is computationally hard since it relies heavily on the factorization of the algebraic norm of f over the integers. In all of the following, for any element $\alpha \in \mathscr{O}_{\mathbf{L}}$, we shall denote by (α) the ideal generated by α in $\mathscr{O}_{\mathbf{L}}$, that is $\alpha\mathscr{O}_{\mathbf{L}}$.

8.2.4 Generalization of Howgrave-Graham-Szydlo algorithm

Let $\mathbf{L}$ be a cyclotomic field of conductor m which is a power-of-two and $f \in \mathbf{L}$, and denote by $\mathbf{L}^+$ its maxima real subfield. We are given the value $f \cdot \overline{f}$ and aim at retrieving f. The algorithm *extracts* the information contained in the relative norm $f \cdot \overline{f}$ by first descending it to the rationals where we can factor it and derive from it the absolute norm of f (**Step I**). Then it lifts all these pieces of information to the base field to yield candidate ideals verifying the same norm equations as the principal ideal $(f) = f\mathscr{O}_{\mathbf{L}}$ (**Step II**). If we get the guarantee that one of them at least is principal, alongside with the possibility to easily retrieve the corresponding generator, then this latter element will be solution of the norm equation (**Step III**). Let us now precise this intuition.

8.2.4.1. ***(Step I)** Norm computation.* The first step aims to compute the norm of the element f over the ground field $\mathbf{Q}$. Since the ideal norm is multiplicative, it corresponds to the square norm of $N_{\mathbf{L}/\mathbf{Q}}(f \cdot \overline{f})$ or equivalently to the absolute norm in the maximal real subfield $\mathbf{L}^+$: $N_{\mathbf{L}^+/\mathbf{Q}}(f \cdot \overline{f})$. Let assume for simplicity that this norm is a -ower of a prime number p from now on, so that $N_{\mathbf{L}/\mathbf{Q}}(f) = p^{\alpha}$. A discussion on the way to adapt the algorithm to the generic case is conducted below. Except from the very specific case of $p = 2$ (ramified case) which is treated separately, two cases occur: either $p \equiv 1 \bmod 4$, or $p \equiv 3 \bmod 4$.

8.2.4.2. ***(Step II)*** *Creation of the candidate ideal(s).* Let study separately what occurs in the two sub-cited cases. The case $p \equiv 1 \bmod 4$ splits itself into two sub-cases, depending on whether $N_{\mathbf{L}/\mathbf{Q}}(f)$ is prime or prime-power.

CASE $p \equiv 1 \bmod 4, \alpha = 1$. One could be surprised by dealing separately with an apparently such restrictive case, but it appears that this case is in fact somehow a generic case. Indeed, the density of prime ideals of norm a strict prime power among all prime ideals is zero. More generally the density of ideals of prime norm among all ideals of prime-power norm is one. Hence, the case $\alpha = 1$ is in this sense generic.

Moreover, if an ideal has an odd prime norm p, then necessarily $p \equiv 1 \bmod 4$. Indeed, if p was congruent to 3 modulo 4, it would be inert in $\mathbf{Z}[i]$. As such the ideal (f) would have a norm over $\mathbf{Z}[i]$ divisible by $p\mathbf{Z}[i]$, meaning that p^2 would divide $N_{\mathbf{L}/\mathbf{Q}}(f) = p$ in $\mathbf{Z}$, yielding a contradiction.

(II-1) SPLIT OF PRIME IN $\mathbf{Q}[i]$. Now that we obtained the norm p, we can consider the principal ideal (p) generated by this prime in the subfield $\mathbf{Q}[i] \subset \mathbf{Q}[\zeta_m]$. The ideal (p) *splits* into two distinct conjugate prime ideals in $\mathbf{Z}[i]$: $(p) = (a + ib) \cdot (a - ib)$, as a consequence of Fermat's theorem[3] on sums of two squares[4].

(II-2) LIFT OF IDEAL. Let consider one of the ideals $(a \pm ib)$ of $\mathbf{Z}[i]$ resulting from the splitting on the quadratic subfield, and *lift* it the whole cyclotomic field, that is seeing it as an ideal of $\mathscr{O}_{\mathbf{L}}$—which is considering the ideals $(a + ib)\mathscr{O}_{\mathbf{L}}$ and $(a - ib)\mathscr{O}_{\mathbf{L}}$. By notational abuse we also denote by $(a + ib)$ (resp. $(a - ib)$) the ideal lifted from $(a + ib)$ (resp. $(a - ib)$).

From these two ideals, we can construct two candidate ideals $\mathfrak{a}_+$ and $\mathfrak{a}_-$ respectively defined as the ideals $(a + ib) + (f \cdot \overline{f})$ and $(a - ib) + (f \cdot \overline{f})$, each of them satisfying the norm equation $\mathfrak{a} \cdot \overline{\mathfrak{a}} = (f \cdot \overline{f})$.

At least one of this candidate ideal is principal and by construction its generator will be solution of the norm equation. The schematic representation of the algorithm in this case is presented in Figure 1.

CASE $p \equiv 1 \bmod 4, \alpha > 1$. For any prime ideal $\mathfrak{P}$ dividing $(f \cdot \overline{f})$, appearing with multiplicity t in its decomposition, $\mathfrak{P}$ can appear in the decomposition of (f) with multiplicity $0 \leq i \leq t$, implying that $\overline{\mathfrak{P}}$ will appear with multiplicity $t - i$ in the decomposition of (f). As such one of the $1 + t$ such ideals appears in the decomposition of (f).

In order to compute (f), we thus need to compute the prime decomposition of $(f \cdot \overline{f})$ and test for the possible multiplicities of each prime ideal of its decomposition. Since we know that the algebraic norm of $f \cdot \overline{f}$ is p^α, we also know that its prime divisors are all primes above p. Then after first enumerating the prime ideals over p, with Berlekamp's algorithm for instance,

3 We let the reader refers to the introductory section of this manuscript for a proof of this theorem using lattices.

4 Indeed, in that case p can be written as the sum of two squares $a^2 + b^2$, yielding directly the announced decomposition.

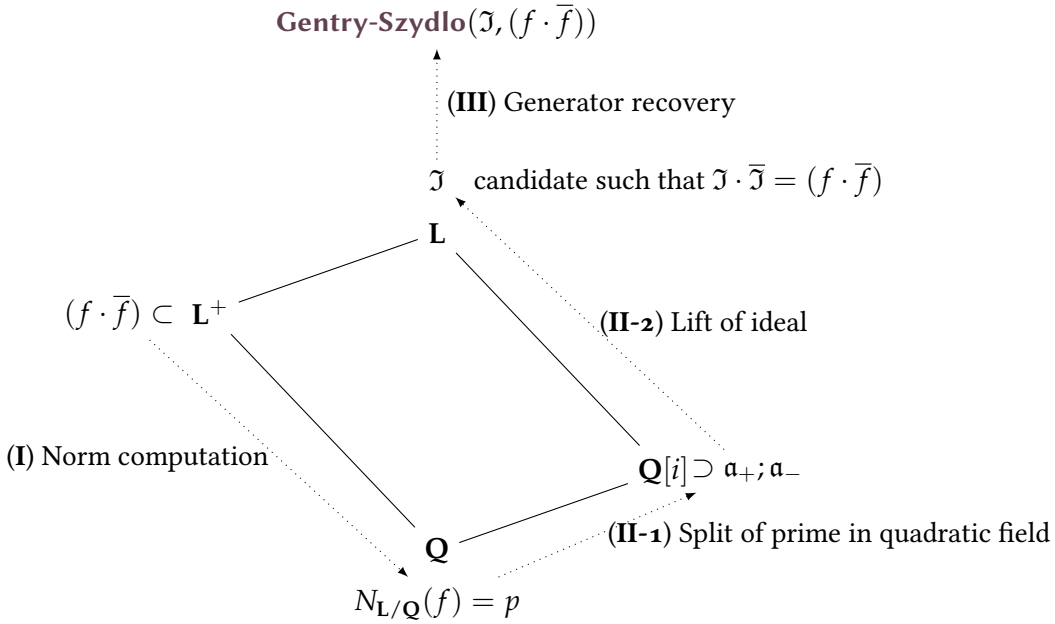


Figure 1: Recovery of the generator f from its relative norm: Case $p \equiv 1 \bmod 4$.

we test the divisibility of $(f \cdot \overline{f})$ by each of their exponentiation. By multiplicativity of the norm each prime ideal can only appear with multiplicity lower than α.

Once the prime decompositions $(f \cdot \overline{f}) = \prod_i \mathfrak{p}_i^{t_i} \overline{(\mathfrak{p}_i)}^{t_i}$ is obtained, we can construct the candidate ideals by computing the products of exactly one ideal of the form $\mathfrak{p}_i^k \overline{(\mathfrak{p}_i)}^{t_i - k}$ among the $1 + t_i$ possibles for each $(\mathfrak{p}_i, \overline{\mathfrak{p}_i})$ pair of conjugate primes, divisors of $(f \cdot \overline{f})$.

CASE $p \equiv 3 \bmod 4$. In this case p is *inert* in $\mathbf{Z}[i]$, and then we can not construct a list of candidate from the decomposition in the quadratic field as in step II-1 of the previous case. Nonetheless this case is somehow simpler: the ideal (f) in $\mathscr{O}_{\mathbf{L}}$ is actually invariant under the conjugation map. Indeed if we decompose (f) in prime ideals: $(f) = \prod_i \mathfrak{p}_i^{e_i}$, each ideal $\mathfrak{p}_i$ is necessarily a real prime ideal over p. Indeed, the norm of each $\mathfrak{p}_i$ over $\mathbf{Q}(i)$ is also real as being a prime over p in $\mathbf{Z}[i]$, that is (p) itself since p is inert. As a consequence, $(f \cdot \overline{f}) = (f)^2$, and we only need to compute the square root of the principal ideal generated by the norm $(f \cdot \overline{f})$ to recover (f). This can be done easily by first decomposing $(f \cdot \overline{f})$ in prime ideals and then dividing the valuation of each prime by two. For notational simplicity and consistence with the ideals generated for the case $p \equiv 1 \bmod 4$, we will also denote the recovered ideal (f) as $\mathfrak{I}$ and call it a candidate ideal. The schematic representation of the algorithm in this case is presented in Figure 2.

8.2.4.3. ***(Step III)*** *Generator recovery.* This final step is now common for the two cases. Given one — or the unique — of the candidate $\mathfrak{I}$ — which is

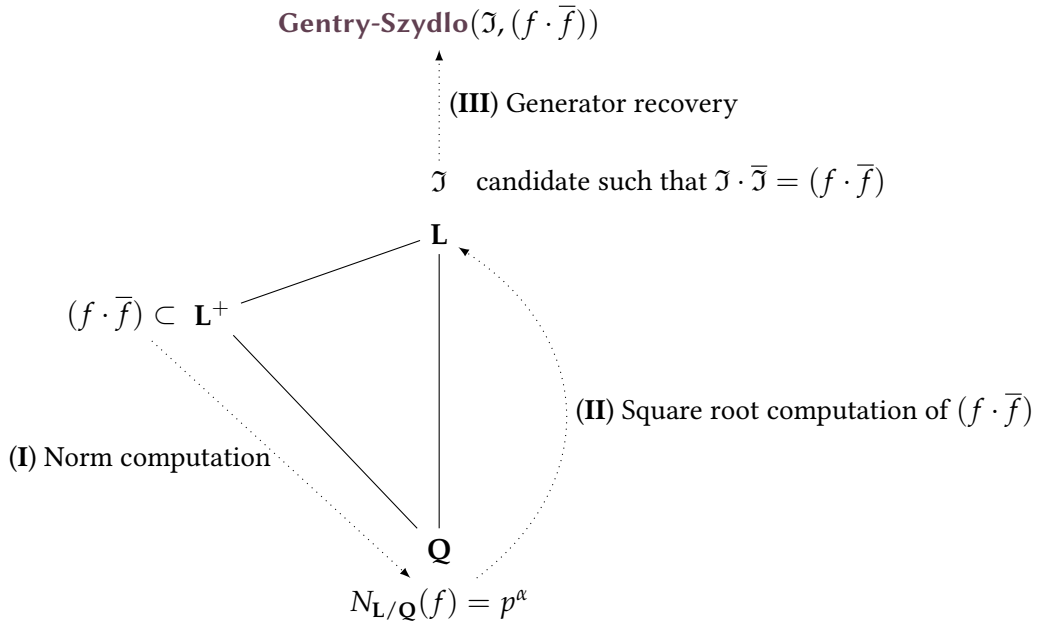


Figure 2: Recovery of the generator f from its relative norm: Case $p \equiv 3 \bmod 4$.

supposed to be principal — as well as the element $f \cdot \overline{f}$, generator of the ideal $\mathfrak{I} \cdot \overline{\mathfrak{I}}$ by construction, the *Gentry-Szydlo* algorithm can be called to recover f up to a root of unity. In the case where $\mathfrak{I}$ is not principal[5], the latter algorithm returns an error, giving hence a method to distinguish principal candidates from others.

8.2.5 The case $p = 2$

The last case is $p = 2$, which is very specific since 2 is the only prime that ramifies in $\mathbf{L}$. It is in fact totally ramified, since

$$\mathscr{O}_{\mathbf{L}} \big/ 2\mathscr{O}_{\mathbf{L}} \cong \mathbf{F}_2[X] \big/ (X^n + 1) \cong \mathbf{F}_2[X] \big/ (X+1)^n,$$

since n is a power-of-two. In particular, the only prime above 2 is $(1 + \zeta_m)$. As such $(f) = (1 + \zeta)^\alpha$, giving directly f up to a unit of the field.

Let us see how to generalize the latter described algorithm in the case of a composite norm $N_{\mathbf{L}/\mathbf{Q}}(f)$.

8.2.6 Composite case

Due to the inherent multiplicative structure of the problem, knowing how to solve it for every element whose norm is a prime power is actually sufficient to solve any instance. Indeed, we perform a reasoning *à la* Chinese reminder theorem: we deal separately with every prime power factor thanks to the study we just carried out and multiplicatively recompose these chunks of solutions.

5 As mentioned, this case can only occur when $p \equiv 1 \bmod 4$.

Thus, the algorithm starts exactly as before, by computing the algebraic norm of the element f, as the square root of the norm of $f \cdot \overline{f}$. In order to deal with every prime factors, we then factor this norm in:

$$N_{\mathbf{L}/\mathbf{Q}}(f) = 2^{\alpha_2} \cdot \prod_i p_i^{\alpha_i} \cdot \prod_j q_j^{\alpha_j},$$

where the $(p_i)_i$ are the prime factors congruent to 1 modulo 4 and the $(q_i)_i$ are the prime factors congruent to 3 modulo 4.

We first take care of the primes $(p_i)_i$ congruent to 1 modulo 4. For primes appearing with multiplicity one, applying the technique described in Section Paragraph 8.2.4.2, each prime p_i yields exactly two ideals above itself divisors of (f), along with the guarantee that at least one of them is principal. As such, since the $(p_i)_i$ are coprimes, we can construct 2^T possible products, for T the number of primes appearing with multiplicity 1, obtained by taking exactly one ideal above each p_i. Let $\mathcal{C}_1$ this set of ideals.

Then, let us turn to the primes appearing with multiplicity greater than one. In order to fall back on the cases described in section Figure 8.2.4.2, we need to construct an ideal of norm $p_i^{\alpha_i}$ dividing the principal ideal generated by the relative norm $(f \cdot \overline{f})$. This is simply the sum of the latter ideal with the principal ideal generated by the element $p_i^{\alpha_i} \in \mathcal{O}_{\mathbf{L}}$. Then applying the technique described in Section Figure 8.2.4.2. each prime p_i yields a certain number c_i of candidate ideals above $p_i^{\alpha_i}$ divisors of (f), along with the guarantee that at least one of them is principal. As such, since the $(p_i)_i$ are coprimes, the possible products obtained by taking exactly one ideal above each $p_i^{\alpha_i}$ and one ideal from the set $\mathcal{C}_1$, are divisors of (f) above $\prod_i p_i^{\alpha_i} = \frac{N_{\mathbf{L}/\mathbf{Q}}(f)}{2^{\alpha_2} \cdot \prod_j q_j^{\alpha_j}}$. At least one of them is principal.

We then treat the case of the primes $(q_j)_j$ congruent to 3 modulo 4. Let q_j one of those primes appearing in the factorization of $N_{\mathbf{L}/\mathbf{Q}}(f)$. In order to fall back on the cases described in section Figure 8.2.4.2, we need to construct a real ideal $\mathfrak{N}$ of norm $q_j^{\alpha_j}$ dividing the principal ideal generated by the relative norm $(f \cdot \overline{f})$. This is simply the sum of the latter ideal with the principal ideal generated by the element $q_j^{\alpha_j} \in \mathbf{Z}[\zeta]$. As in Figure 8.2.4.2 we construct a principal $\mathfrak{I}_{q_j}$ of norm $q_j^{\alpha_j}$ dividing (f). Performing this construction on every prime q_j and denoting by $\mathfrak{R}$ the product of each freshly obtained $\mathfrak{I}_{q_j}$, ensures that the principal ideal $\mathfrak{R}$ is a divisor of the ideal (f) above $\prod_j q_j^{\alpha_j} = \frac{N_{\mathbf{L}/\mathbf{Q}}(f)}{2^{\alpha_2} \cdot \prod_i p_i^{\alpha_i}}$.

Finally, we deal with the power-of-two appearing in the norm. The reasoning is similar to what happens in Section Section 8.2.5: the prime power principal ideal $\mathfrak{D} = (1 + \zeta_m)^{\alpha_2}$ is a divisor of (f) above 2^{α_2}.

It is now time to reconstruct candidate ideals from these three parts. Multiplying each of the $\prod_i (1 + \alpha_i)$ candidates obtained from the $(p_i)_i$ with the principal ideal $\mathfrak{R} \cdot \mathfrak{D}$ yields a candidate ideal of norm $N_{\mathbf{L}/\mathbf{Q}}(f)$. Eventually, taking the sum with the ideal $(f \cdot \overline{f})$ gives then a list of $\prod_i (1 + \alpha_i)$ ideals satisfying the norm equation, with the guarantee that at least one of them is principal. The final step of the algorithm is then unchanged: finding the gen-

erator of the principal ideal by using the Gentry-Szydlo algorithm. A method of reducing the running time of this final phase is to process all possible candidate ideals in parallel and stop as soon as one of the process returns a generator. The full outline of the algorithm is given in Algorithm 31.

Algorithm 31 — Generalized Howgrave-Graham–Szydlo

Input : Relative norm $f \cdot \overline{f}$.
Output : Algebraic integer α such that $\alpha \cdot \overline{\alpha} = f \cdot \overline{f}$.

Compute the norm $N_{\mathbf{L}/\mathbf{Q}}(f)$ as $\sqrt{N_{\mathbf{L}/\mathbf{Q}}(f \cdot \overline{f})}$
Factor $N_{\mathbf{L}/\mathbf{Q}}(f)$ in prime product $\prod_i p_i^{\alpha_i}$
for ***each*** p_i *such that* $p \equiv 3 \ mod\, 4$ **do**
  Split the ideal $(f \cdot \overline{f}) + (p_i^{\alpha_i})$ of $\mathscr{O}_{\mathbf{L}}$ in primes: $\prod_j \mathfrak{p}^{e_j}$
  $\mathfrak{I}_{p_i} \leftarrow \prod_j \mathfrak{p}^{e_j/2}$
end for
$\mathfrak{R} \leftarrow \prod_{p_i \equiv 3[4]} \mathfrak{I}_{p_i}$
$\mathfrak{D} \leftarrow (1+\zeta)^{\alpha_2}$
for ***each*** p_i *such that* $p \equiv 1 \ mod\, 4$ *and* $\alpha_i = 1$ **do**
  Split (p_i) in $\mathbf{Z}[i]$ as $(a+ib)$, $(a-ib)$
  Lift $(a+ib)$, $(a-ib)$in $\mathbf{Z}[\zeta]$
  $\mathfrak{a}_+ \leftarrow (a+ib) + (f \cdot \overline{f})$
  $\mathfrak{a}_- \leftarrow (a-ib) + (f \cdot \overline{f})$
  $\mathcal{C}_{p_i} \leftarrow \{\mathfrak{a}_+, \mathfrak{a}_-\}$
end for
for p_i *such that* $p \equiv 1 \ mod\, 4$ *and* $\alpha_i > 1$ **do**
  Factor $(f \cdot \overline{f}) + (p_i^{\alpha_i})$ as $\prod_j \mathfrak{p}_j^{t_j}$
  for ***each*** $(\mathfrak{p}_j, \overline{\mathfrak{p}_j})$ *of the decomposition* **do**
    $t_j \leftarrow$ multiplicity of $\mathfrak{p}_j$ in $(f \cdot \overline{f})$
    $\mathcal{C}_{p_i,\mathfrak{p}_j} \leftarrow \left\{\mathfrak{p}_j^{t_j}, \mathfrak{p}_j^{t_j-1}\overline{\mathfrak{p}_j} \ldots, \mathfrak{p}_j^{k}\overline{\mathfrak{p}_j}^{t_j-k}, \ldots, \overline{\mathfrak{p}_j}^{t_j}\right\}$
  end for
  $\mathcal{C}_{p_i} \leftarrow \left\{ \prod_{\mathfrak{b} \in B} \mathfrak{b} \quad | \quad B \in \prod_j \mathcal{C}_{p_i,\mathfrak{p}_j} \right\}$
end for
for $A \in \prod_{p_i} \mathcal{C}_{p_i}$ **do**
  $\mathfrak{I} \leftarrow \mathfrak{R}\mathfrak{D} \cdot \prod_{\mathfrak{b} \in A} \mathfrak{b}$
  if ***Gentry-Szydlo***$(\mathfrak{I}, f \cdot \overline{f})$ *outputs* $\alpha \in \mathbf{Z}[\zeta]$ **then**
    return α
  end if
end for

8.2.6.1. *Remarks on complexity.* Performing operations on ideals in a number field of degree n is polynomial in n, since when working with HNF representation of ideals, the computations of sum, product or intersections of ideals boils down to basic linear algebra computations and calls to an HNF oracle, which is known to be polynomial in the dimension (see for instance

Chapter 3 to 5 of [34] for a complete introduction to computations with ideals). As early mentioned by Dedekind in [43], computing the decomposition in prime ideals of a given prime[6] boils down to factor the defining polynomial of the field Φ_m modulo p, which can be efficiently performed by the Cantor-Zassenhaus algorithm or Berlekamp algorithm [59].

8.3 YET ANOTHER FULL-KEY RECOVERY ON BLISS

In the previous section, we exploited the leakage coning from the computation of the relative norm of the secret, coming from the Bernoulli sampler of exponential parameter. It appears that even if we patch this sampler to be constant-time (by going through the whole loop instead of early aborting) and try to avoid the algebraic attack, the second Bernoulli sampler, of hyperbolic cosine parameter, is leaking.

Indeed, by definition of the function **SampleBernCosh**, the probability of outputting a is equal to the probability of the expression $\neg a \wedge (b \vee c)$ to be false, which is

$$\begin{aligned} p(S,c,z) &= 1 - \Pr(\neg a)\Pr(b \vee c) \\ &= 1 - (1 - \Pr(a))(1 - \Pr(\neg b \wedge \neg c)) \\ &= 1 - \left(1 - e^{-\frac{|\langle z, Sc\rangle|}{2\sigma^2}}\right)\left(1 - \frac{1 - e^{-\frac{|\langle z, Sc\rangle|}{2\sigma^2}}}{2}\right) = \frac{1 + e^{-\frac{|\langle z, Sc\rangle|}{\sigma^2}}}{2}. \end{aligned}$$

Therefore, by measuring the differences in computation time, one can derive traces that shape (z, c, t), where $t \in \mathbf{N}$ is the number of restarts performed before outputting the value a. In the following of this section, we describe two ways to exploit this leakage, leading again to a full key recovery.

8.3.1 Spectral attack with samples with $t = 0$

Remark that if a trace satisfies $t = 0$, then it is likely for the geometric distribution parameter $p(S, c, z)$ to be large. Therefore, for such a sample, $\langle z, Sc\rangle$ should be close to zero, i.e., S should be *close to* be orthogonal to the vector zc^*, where c^* is the adjoint of c: $\langle z, Sc\rangle = \langle zc^*, S\rangle$.

If the vector S was actually orthogonal to each of these zc^* then it would be enough to collect sufficiently of them so that they generate an hyperplane $\mathcal{H}$ of the ambient space $\mathbf{R}^n$ and return the unique (up to sign) vector of $\mathcal{H}^\perp$ of norm compatible with the specification of BLISS (secret vectors in BLISS all have the same known norm by construction). This would practically translate in constructing the empirical covariance matrix $W = \sum_i w_i w_i^T$

6 In full generality, this is the case only when p does not divide the index $[\mathcal{O}_\mathbf{K} : \mathbf{Z}[\alpha]]$ for $\mathcal{O}_\mathbf{K}$ the ring of integers of $\mathbf{K}$ and α a primitive element of the number field $\mathbf{K}$. Since this index is always 1 for cyclotomic fields, the factorization can always be carried by the above-mentioned technique.

$(w_i = z_i c_i^*)$ for a series of trace $(z_i, c_i, 0)$ and get a basis of its kernel. Remark now that since the secret is not actually orthogonal to these vectors, the obtained matrix is not singular. To overcome this difficulty we thus do not seek a vector in the kernel but instead in the eigenspace associated with the smallest eigenvalue of W. This technique can be seen as a continuous relaxation of the kernel computation in the prefect case. It translates directly into pseudocode in Algorithm 32, where the computation of the eigenvector is performed iteratively and $N = \lceil \delta_1 n \rceil + 4\lceil \delta_2 n \rceil$ is the norm of the secret key. Remark that this technique does not recover exactly the secret but an *approximate* solution over the reals. To recover the secret we need to find the closest integral vector to the output candidate, which is simply done by rounding each coefficient to the nearest integral elements. In addition, remark that by the contruction of the public key from the secret one, recovering solely s_2 is sufficient to reconstruct the full secret key. Hence the rounding can be carried to $2\mathbf{Z}$ on the second part of the eigenvector to conclude, as s_2 has its coefficients equal to 0, ± 2 or ± 4 by construction.

Algorithm 32 — Spectral attack

Collect m traces (z_i, c_i, t_i)
for $i = 0$ ***to*** m **do**
  if $t_i = 0$ **then**
    $W \leftarrow c_i z_i^* \cdot c_i {z_i^*}^T$
  end if
end for
$S \leftarrow_{\$} \mathcal{N}(0,1)^n; S \leftarrow \frac{S}{\|s_0\|}$
for $i = 0$ ***to*** K **do**
  $S \leftarrow W^{-1} S; S \leftarrow \frac{S}{\|s_0\|}$
end for
return round$(\frac{S}{\|S\| N})$

Let us now try to use *all* the samples and not discard the ones for which $t > 0$. j

8.3.2 A timing attack by phase retrieval

Exploiting the leakage of the sampler boils down to retrieve S up to sign from a family of values of the shape (z_i, c_i, t_i) where t_i is sampled under a *geometric* distribution of parameter $p(S, c_i, z_i)$. A natural approach would then consist in starting by estimating the values of $p(S, c_i, z_i)$ for each trace (c_i, z_i, t_i), yielding a (noisy) estimate of the absolute value of the inner product $|\langle z_i, S c_i \rangle| = |\langle z_i c_i^*, S \rangle|$. In a second time we then fall back on retrieving S from samples of the form $(|\langle w_i, S \rangle|, w_i)$. This is an instance of so-called (noisy) *phase retrieval problem.*

8.3.2.1. *First phase: estimation of the phases.* In order to get a (noisy) evaluation of the phases, we devise an estimator of maximum likelihood. For any x, set $\mathcal{L}_i(\omega)(x)$ to be the logarithm of the probability $\Pr[|\langle S, w_i\rangle| = x | t = \omega]$. We then set the estimator y_i to be the arguments of the maximum of $\mathcal{L}_i(t_i)$ for each trace. Such a computation is classically done using Bayes' theorem and seeking for critical values from the derivates of $x \mapsto \mathcal{L}_i(\omega)(x)$.

8.3.2.2. *Second phase: solving the phase retrieval instance.* Phase retrieval aims at solving quadratic equations of the shape

$$|\langle S, w_i\rangle|^2 = y_i \quad i = 1, \ldots, m,$$

where S is the decision variable, the w_i are known sampling vectors and the $y_i \in \mathbf{R}$ are the phase measurements. The noisy version of this problem consists in retrieving the variable S from noisy quadratic equations:

$$|\langle S, w_i\rangle|^2 + e_i = y_i \quad i = 1, \ldots, m,$$

for e_i independents (usually gaussian) random variables. This non-noisy problem has been widely studied in the fields of statistical learning and the most common approach to tackle it consists of a two-step strategy. It appears that with some minor tweaks, this method is robust enough to tackle our noisy version.

8.3.2.3. *Initialization via spectral method.* First, find a candidate vector s_0 that is sufficiently close to the actual solution to make the second step converges towards the actual solution. The usual way to initialize the candidate vector can be seen as a generalization of the principal component analysis (PCA): the initial guess is given via a spectral method; in short, s_0 is the leading eigenvector of the positive definite symmetric matrix $\sum_i y_i w_i w_i^T$. The intuition behind this method is to remark that the secret vector will have a greater inner product with the test vectors w_i which have a small angle with it. Hence we want to extract the direction of the w_i for which the inner product is the largest, that is, favoring the components inducing high y_i's. This corresponds to extract the largest eigenvalue of the Gram-matrix of the w_i, normalized by a diagonal matrix of y_i. It is nothing more than a principal component analysis on the test vectors w_i. In practice, we use a slightly different version of the (iterative version of the) spectral initializer, outlined in Algorithm 33, which provides slightly better[7] practical results than the classical method of [31]. In this algorithm, $\mathcal{N}(0,1)$ is the centered normal reduced distribution, K is a constant, set sufficiently large and N is the (public) norm of the secret key of the BLISS signature.

7 Interestingly, our initializer outperforms the state-of-the-art techniques for a wide range of statistical datasets. A precise theoretical analysis of this algorithm would be interesting for its own sake and is a future research topic we desire to tackle.

Algorithm 33 — Spectral Initializer

Input : Test vectors $(w_i)_i$ and their corresponding measures $(y_i)_i$
Output : A candidate solution s_0

$A \leftarrow [w_1 \mid \cdots \mid w_m]$
$s_0 \leftarrow_\$ \mathcal{N}(0,1)^n$
for $i = 0$ ***to*** K **do**
  $s_0 \leftarrow A^T \mathrm{diag}(y_1, \ldots, y_m) A s_0$
  $s_0 \leftarrow (A^T A)^{-1} s_0$
  $s_0 \leftarrow \frac{s_0}{\|s_0\|}$
end for
$s_0 \leftarrow \frac{s_0}{\|s_0\|} N$
return rounding(s_0)

8.3.2.4. *The descent phase.* Once an initialization vector is found, we iteratively try to make it closer to the actual secret by a series of updates like in a gradient descent scheme. Note that in the problem of phase retrieval the problem is non-convex so that a direct gradient descent would not be directly applicable. As stated in [31], the phase retrieval problem can be stated as a minimization problem:

$$\text{minimize} \quad \frac{1}{2m} \sum_{r=1}^{m} \ell(y_r, |\langle w_r, x \rangle|^2), z \in \mathbf{R}^n, \tag{8.2}$$

where ℓ is a distance function over the reals (such as the Euclidean distance $\ell_2(a,b) = (a-b)^2$). The corresponding descent, called Wirtinger flow, is then simply stated in Algorithm 34 where $t \mapsto \mu_t$ is a step function, which has to be experimentally tailored to optimize the convergence. The value $\epsilon > 0$ is a small constant that determines the desired precision of the solution.

It is well known that minimizing non-convex objectives, which may have very many stationary points is in general NP-hard. Nonetheless if the initialization s_0 is sufficiently accurate, then the sequence s_i will converge toward a solution to the problem given by Equation 8.2.

As in the first attack, the descent algorithm does not directly give an integral solution to the retrieval problem, so that we eventually need to round the coefficients before outputting the solution.

The full outline of the attack is given in Algorithm 35.

Algorithm 34 — Wirtinger Flow Descent

Input : Test vectors $(w_i)_i$ and their corresponding measures $(y_i)_i$, initial guess s_0
Output : A candidate solution s_0

$t \leftarrow 0$
do
  $s_{t+1} \leftarrow s_t - \frac{\mu_t}{m\|s_0\|^2} \sum_{r=1}^{m} (|\langle w_r, s_t \rangle|^2 - y_r)(w_r w_r^t) s_t$
  $t \leftarrow t + 1$
while $\|s_t - s_{t+1}\| > \epsilon$
return Rounding$S$

Algorithm 35 — Full timing attack

Collect m traces (z_i, c_i, t_i)
for $i = 0$ **to** m **do**
  $y_i \leftarrow (\text{argmax}_x \mathcal{L}_i(t_i)(x))^2$
end for
$s_0 \leftarrow$ **Spectral Initializer**$((z_i c_i^*)_i, (y_i)_i)$
$S \leftarrow$ **Wirtinger flow Descent**$((z_i c_i^*)_i, (y_i)_i, s_0)$
return S

8.3.3 Reducing the number of samples by error localization and dimension reduction

By the inherent noisy nature of the problem, if not enough samples are used to mount the attack, the recovery might fail on a certain amount of bits. In such a case one cannot figure *a priori* where these errors are and would be forced to enumerate the possible errors, using, for instance, the hybrid MiTM technique of Howgrave-Graham [86]. Since the dimension ($n = 512$) is large, such an approach becomes quickly untractable as the number of errors is greater than 8.

However, as the final step of both of the attacks consists of a coefficient-wise rounding, we can study the distance of each coefficient to $2\mathbf{Z}$. Heuristically since the descent is supposed ultimately to converge to the secret, the retrieved coefficients should be close to $2\mathbf{Z}$. Hence if some of them are far from this lattice, we can consider them as problematic coefficients and likely to be prone to induce an error after rounding. Suppose that we discriminate these problematic coefficients in a finite set T and that each coefficient outside T is correctly retrieved by rounding. Then we can find the correct value of the coefficients in T by lattice reduction in dimension slightly larger than $|T|$ by the exploitation of *dimension reduction techniques* we introduced j in [52].

If the resulting dimension with this reduction is sufficiently small (less than 100 for typical computers), this approach allows to still perform a full

key recovery in cases where the sole descent algorithm would have led to some errors.

8.3.4 Practicality of the attacks and discussion

We summarize in Table 8.4 the number of samples required to perform a full key recovery with both of the attacks. The first column corresponds to the first attack described in Section 8.3.1 with the MiTM technique of [86] to correct the errors. The second column corresponds the Wirtinger flow technique coupled with the lattice reduction and the localization of Section 8.3.3. Since the descent attack is an improvement build on a spectral method, it is natural to see that this algorithm indeed requires far fewer samples to mount the attack than the first method presented in Section 8.3.1. It should also be noticed that this attack discards every samples for which $t > 0$, implying that a certain amount of the information provided by the samples is not used. For instance when attacking BLISS-II with compression, almost 30 millions of samples are necessary to retrieve the secret, but among those, only 18 millions of them are actually conserved to mount the attack. The number of required samples may seems high compared to the dimension of the problem, but it can be noticed that the size of the errors obtained by obtaining the estimation of the phases by maximum likelihood is of the same magnitude as the actual phase we are trying to retrieve. Hence, canceling the noise actually costs a significant amount of samples, as evoked above.

As far as the correction of errors is concerned, with the two techniques introduced in Section 8.3.3 (i.e. the MiTM and the localization), the two attacks have different behaviors. Indeed, the MiTM exhaustive search appeared to be more tailored to the first attack whereas the localization worked far better for the descent attack. A more detailed discussion on the causes of this phenomena is provided in Section 8.3.5 below. The results presented in Table 8.4 are obtained by making the maximum use of these correction techniques. Hence, the running time of a full key recovery is contributed almost exclusively by this final phase: practically the parameters given allows the descent to yield a lattice problem in dimension at most 110. On a Intel Xeon E5-2697v3 workstation this phase takes less than an hour to complete. Using the DBKZ reduction with blocksize 25 takes then around 38h to complete the recovery.

A striking observation is that in both of the attacks the compression on z_2 used in actual BLISS signatures, makes the recovery significantly harder: indeed, there is an order of magnitude between the number of samples needed to make a full key recovery. Indeed the bit dropping yields noisier estimates for the recovery problem. Finally, note that BLISS-II is the hardest variant to attack with this method. This is due to the fact that this parameter set provides the highest rate of compression.

Table 8.4: Experimental number of samples required to perform a full key recovery. The average CPU time for a full key recovery is 40h on a Intel Xeon E5-2697v3 workstation.

		PCA_0+MiTM	Spectral+Descent
w/o compress	BLISS-I	180k	65k
	BLISS-II	250k	130k
	BLISS-III	209k	100k
	BLISS-VI	308k	120k
w/ compress	BLISS-I	4200k	700k
	BLISS-II	27500k	2000k
	BLISS-III	2100k	350k
	BLISS-VI	unfeasible	200k

8.3.5 Convergence behavior

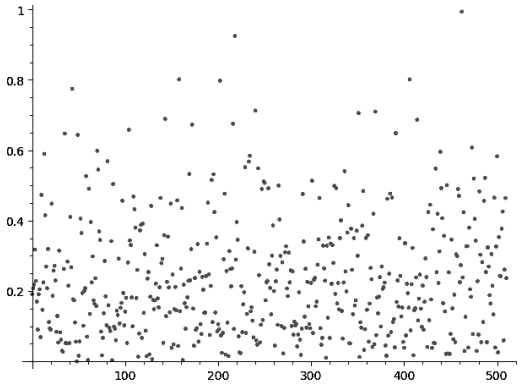
In Figure 3 we present the result of an experiment picturing the distance of each coefficient of the candidate secret from the lattice $2\mathbf{Z}$ before the final rounding, for both of the proposed attacks.

A striking observation is that the descent attack pushes way more the distances towards either 0 or 1 and as such makes it easy to localize the coefficients that are prone to be problematic. Indeed setting a threshold at 0.5 clearly discriminates the "good" coefficients form the potentially problematic ones. On the contrary, the situation is way more blurry in the other attack, where the distances are much more close to 0.5. As such being able to distinguish the "good" coefficients from the "bad" ones is much more difficult in order not to create false positives.

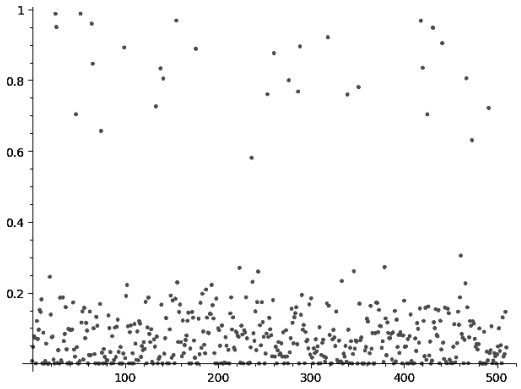
As a consequence, it is experimentally less costly to rely on Meet in the Middle technique to resolve the errors in this latter case as setting a threshold too low would imply reducing lattices of dimension too large.

8.4 TOWARDS A CONSTANT-TIME IMPLEMENTATION

The main feature of BLISS exploited in the attack presented and in some other exploitation of side-channel traces (see for instance [20, 24, 134]), is the use of discrete Gaussian distributions, either as part of the Gaussian sampling used to generate the random nonces in BLISS signatures, or as part of the



(a) Eigenvalue retrieval



(b) Descent technique

Figure 3: Comparison of the repartition of the distance to the lattice $2\mathbf{Z}$.

crucial rejection sampling step that forms the core of the Fiat–Shamir with aborts framework that supports BLISS's security.

Generally speaking, Gaussian distributions are ubiquitous in theoretical works on lattice-based cryptography, thanks to their convenient behavior with respect to proofs of security and parameter choices. However, their role in practical implementations is less clear, largely because of the concerns surrounding implementation attacks. Hence it seems important to develop efficient implementations of BLISS that are secure against these attacks, and in particular of the Gaussian generation part.

A possible solution can be find by relying on an alternate implementation of the rejection sampling step, carried out by computing a sufficiently precise polynomial approximation of the rejection probability using pure integer arithmetic. We manage to do so using a novel technique for polynomial approximation in Sobolev norm[8]. As this norm is Euclidean, we can solve the polynomial approximation problem using the techniques introduced in Chapter 4 to reduce the corresponding $\mathbf{Z}$-module endowed with the (real-

8 For u and v two differentiable functions defined on an interval I, Sobolev H^2 inner product is defined by the following

$$\langle u, v \rangle = \int_I uv + u'v'$$

valued) inner product coming from this norm. We do not detail the precise construction here as it get out of the scope of this manuscript but still remark that the techniques developed in this thesis can also be applied for constructive matters in post-quantum cryptography. For a detailed reference on this constant time-implementation, we let the interested reader refer to our work [9].

The corresponding norm $\|\cdot\|_{H^2}$ is

$$\|u\|_{H^2} = \sqrt{\int_I (u^2 + u'^2)}$$

.

9

CONCLUDING REMARKS AND OPEN PROBLEMS

All along this manuscript, we tried to reveal relations between lattices and arithmetic, through the prism of the algorithmic of lattice reduction. In particular, we highlighted the notion of algebraic lattices and showcased some applications of the reduction of these objects in number theory and in cryptography. Our objectives were twofold: on the one hand we aimed at providing a sound framework to perform the reduction of arbitrary real lattices—and in particular to make possible the certified reduction of algebraic lattices by using their image over $\mathbf{Z}$—, and on the other hand we wanted to avoid the dimensional blow-up coming from the descent over $\mathbf{Z}$ to speed up such reduction, by exploiting the algebraic structure of these lattices.

9.1 GENERALIZING THE REDUCTION TO ANY NUMBER FIELDS

However, the reduction techniques introduced in Chapter 5, lead to a *heuristic* reduction of algebraic lattices over cyclotomic fields. This reduction was focusing on the reduction of *free* algebraic lattices. From this first step, we can now aim at extending the reduction to arbitrary projective modules and thus to arbitrary algebraic lattices. This requires to move from bases to pseudo-bases. The whole resulting reduction is actually similar to the one presented in this manuscript, but the lifting phase is presenting the most differences as it now acts on ideals. We need to generalize the Euclidean algorithm for solving Bezout equation to the ideal setting. As a consequence, we also need to compute with ideals at each step of the reduction. A naive approach to ideal arithmetic would become the bottleneck of the reduction. However, using the idea introduced in Section 1.5.3, we can obtain fast arithmetic if we can fastly compute a two-element representation from multiple generators. This computation can be done with lattice reduction. This gives a two-way recursive procedure: on the one hand, the reduction itself makes calls to a fast arithmetic procedure which itself may require to reduce some algebraic lattices. Besides, this reduction is fully provable and works with any tower of number fields: it is not restricted to the cyclotomics fields. Besides the fast ideal arithmetic, an application of this reduction is the fast computation of the Hermite Normal Form for modules over number fields.

As the rational arithmetic is performed using floating-point in the evoked algorithm, a natural line of research is then to combine this algorithm with the interval arithmetic presented in Chapter 4 in order to certify the reduction while using minimal precision.

9.2 TOWARDS A BLOCKWISE REDUCTION OF ALGEBRAIC LATTICES?

A natural question remains open from the research presented in this manuscript and even from the generalization we evoked: how does this algebraic framework generalizes to the blockwise reduction algorithms, à la BKZ? Indeed, we have seen in Chapter 2 that a generic way to improve the quality of reduction is to adjoin a SVP-oracle to the reduction. Therefore a faster SVP-oracle for algebraic lattices would allow designing a blockwise reduction tailored for algebraic lattices, replacing the LLL reduction by the fast reduction of Chapter 5. However, up-to-our knowledge, no significant improvement for the exact-SVP problem for algebraic lattices exists. Trying to exploit the symmetries and algebraic structure of number fields to construct ad-hoc enumeration or sieving processes constitutes a thriving open problem for enhancing the reduction of algebraic lattices. Implications in lattice-based cryptography would be of course deep as it would force to reevaluate the security of all primitives using algebraic lattices.

9.3 TOWARDS N-PLECTIC REDUCTION

In Chapter 5, we introduced an effective technique to make compatible the symplectic structure with the direct image of a lattice over a subfield. This allowed attaching a symplectic structure to an algebraic lattice over a tower of number fields. This technique allowed to greatly improve the efficiency of lattice reduction for algebraic lattices. Indeed, recall that, informally, symplectic geometry aims at studying transformations letting areas between pair of vectors unchanged. Hence, by using only specific transformations to manipulate a symplectic lattice we could preserve its symplectic structure: each local transformation on a basis let invariant a specific area element so that acting on a vector also modifies another vector. All in all, it suffices to reduce only half of the basis to act on the whole basis by the preservation of the symplectic structure.

However, the drawback of this approach is to make the approximation factor of the reduction slightly worse. But then, instead of looking at transformations preserving areas, we could try to use transformations preserving volume forms or even more generally $n-$volume forms. Instead of relying on *symplectic* geometry, we would now enter the realm of $n-$*plectic geometry*, or so-called *higher-order symplectic* geometry. In this case, a local transformation of $n-1$ vectors of a basis would suffice to transform n vectors. The resulting reduction would, of course, be slower than with the symplectic technique used, but the approximation factor would be improved.

If the transition from lattice reduction to symplectic lattice reduction is pretty straightforward, going to $n-$plectic reduction seems to us a bit trickier. Indeed, a nice property of the symplectic spaces (or more generally symplectic manifolds) is that there is essentially a unique symplectic structure, by Darboux theorem (the structure being given by any Darboux basis, as explained in Chapter 5. This is no longer the case for n-plectic geometry. In

particular, there is no direct correspondence between a "size-reduction" and a discretized version of the computation of a *canonical* basis of a $n-$plectic space. Hopefully, for the volume form coming from the Euclidean structure, it appears that we can adapt the construction of such a basis to overcome this obstruction. However, finding a descent of this structure which remains compatible with the direct image is still an open and very interesting problem.

9.4 APPLICATION TO COMPUTATIONAL ARAKELOV THEORY

Arakelov theory is a geometric approach to Diophantine equations and approximation in high dimensions. However, very few works have been done in the design of efficient algorithms in algebraic Arakelov theory and in particular in the computation of the Arakelov class group of a number field. An important result comes from the original remark of Schoof in [149], where he noticed that the infrastructure machinery used by Shanks [151] for the computations of class group à la Buchmann [25] are very naturally described within the Arakalov class group of a number field. This group consists of the quotient of so-called Arakelov divisors by principal Arakelov divisors, which are in correspondence with line bundles over the arithmetic curve given by spectrum of a maximal order of the field. Hence computations in this setting *require the reduction of vector bundles over such curves*, which can be done *efficiently and in a certified manner* by developing the lattice reduction techniques introduced in Chapter 4 and Chapter 5. Hence the techniques introduced in this thesis are naturally tailored for handling the base objects appearing in this theory and form a first (baby) step towards effectiveness in this theory.

Conversely, a natural investigation would be to make the Arakelov machinery revolving around the study of vector bundles over curves to *irrigate the analysis and design of lattice reduction alogrithms*. We have seen in Chapter 2 of numbers that reductions algorithms can be interpreted as acting on filtrations of lattices. Hence they can be thought as a greedy approximation of the *Harder-Narashiman filtration* [76] of the corresponding vector bundle. On the one hand, viewing lattice reduction as acting on such approximate stratification gives a new way to *understand and analyze classical algorithms* whereas on the other hand, trying to find efficient algorithms to approximate the Harder-Narashiman filtration would make arise *new reduction techniques* for Euclidean or algebraic lattices. As a very basic instance of this program, it seems to us the filtration machinery is a more natural mathematical tool to describe and analyze lattice reduction, compared to the usual way of dealing with Gram-Schmidt vectors. Of course, any result can be proved by one or the other approach, but it appears the computations can be done in a cleaner manner using filtrations.

9.5 GENERALIZATION OF THE LWE-LIKE PROBLEMS TO AN ALGEBRO-GEOMETRIC SETTING

The celebrated LWE problem of Regev [140] and its numerous variants (Ring LWE, Module LWE, Order LWE, Middle-Product LWE, Torus LWE, among others) all live in a common natural arithmetic-geometric setting. Indeed these problems can be seen as *learning a morphism from a vector bundle over an arithmetic curve to a compact variety*—the classical LWE problem being encompassed by the trivial vector bundle over Spec $\mathbf{Z}$ to the real torus $\mathbf{R}/\mathbf{Z}$—. This general setting would allow *simplifying and uniting* the *numerous* variants and proofs, as well as being able to use the whole arsenal of arithmetic geometry to refine and develop the theory revolving around these problems. In particular, we can wonder if it is possible to give less ad-hoc proofs of hardness and rely more on the *interaction between the geometry of the problem and its information theoretical analysis*. Moreover, a non-commutative version of Arakelov theory has been proposed [21] to study bundles over the spectrum of a $\mathbf{Z}$-order in finite-dimensional semisimple $\mathbf{Q}$-algebras. The formalism evoked above would then be extendible to non-commutative curves and would lead to a very generic LWE-like problem in a non-commutative geometric setting. Cryptographic constructions arising from this formalism would keep the efficiency of lattice-based cryptography while being less sensible to linear representation-based attacks. A question which arises then is the *possibility to construct secure graded-encoding schemes* from such non-commutative learning problems—which would be resistant to the disastrous zeroizing attacks [39] breaking all of the previous constructions of such schemes—.

BIBLIOGRAPHY

[1] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. "Solving the shortest vector problem in 2^n time using discrete Gaussian sampling: extended abstract." In: *47th Annual ACM Symposium on Theory of Computing*. Ed. by Rocco A. Servedio and Ronitt Rubinfeld. Portland, OR, USA: ACM Press, 2015, pp. 733–742.

[2] Miklós Ajtai. "Generating hard instances of lattice problems." In: *28th Annual ACM Symposium on Theory of Computing*. Philadephia, PA, USA: ACM Press, 1996, pp. 99–108.

[3] Sedat Akleylek, Nina Bindel, Johannes A. Buchmann, Juliane Krämer, and Giorgia Azzurra Marson. "An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation." In: *AFRICACRYPT*. Ed. by David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi. Vol. 9646. LNCS. Springer, 2016, pp. 44–60.

[4] Yves André. "On nef and semistable hermitian lattices, and their behaviour under tensor product." In: *Tohoku Mathematical Journal, Second Series* 63.4 (2011), pp. 629–649.

[5] Yoshinori Aono, Thomas Espitau, and Phong Nguyen. "Random lattices: theory and practice." Available at http://www.espitau.github.io/. 2019.

[6] Yoshinori Aono and Phong Q. Nguyen. "Random sampling revisited: lattice enumeration with discrete pruning." In: *Advances in Cryptology – EUROCRYPT 2017, Part II*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. Lecture Notes in Computer Science. Paris, France: Springer, Heidelberg, Germany, 2017, pp. 65–102.

[7] Yoshinori Aono, Phong Q. Nguyen, Takenobu Seito, and Junji Shikata. "Lower Bounds on Lattice Enumeration with Extreme Pruning." In: *Advances in Cryptology – CRYPTO 2018, Part II*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10992. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2018, pp. 608–637.

[8] Ram Prakash Bambah, Alan Woods, and Hans Zassenhaus. "Three proofs of Minkowski's second inequality in the geometry of numbers." In: *Journal of the Australian Mathematical Society* 5.4 (1965), pp. 453–462.

[9] Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi, and Mehdi Tibouchi. "GALACTICS: gaussian sampling for lattice-based constant-time implementation of cryptographic signatures, revisited." In: (2019).

[10] Christian Batut, Karim Belabas, Dominique Bernardi, Henri Cohen, and M Olivier. "Pari-GP." In: *Avaliable from ftp://megrez. math. u-bordeaux. fr/pub/pari* (1998).

[11] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. "New directions in nearest neighbor searching with applications to lattice sieving." In: *27th Annual ACM-SIAM Symposium on Discrete Algorithms*. Ed. by Robert Krauthgamer. Arlington, VA, USA: ACM-SIAM, 2016, pp. 10–24.

[12] Karim Belabas. "Topics in computational algebraic number theory." In: *J. Théor. Nombres Bordeaux* 16 (2004), pp. 19–63.

[13] Jean-François Biasse. "Subexponential time relations in the class group of large degree number fields." In: *Advances in Mathematics of Communications* 8.4 (2014), pp. 407–425.

[14] Jean-François Biasse and Claus Fieker. "Subexponential class group and unit group computation in large degree number fields." In: *LMS Journal of Computation and Mathematics* 17 (2014), pp. 385–403.

[15] Jean-François Biasse and Claus Fieker. "A polynomial time algorithm for computing the HNF of a module over the integers of a number field." In: *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*. ACM. 2012, pp. 75–82.

[16] Jean-François Biasse and Fang Song. "Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields." In: *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms SODA 2016*. 2016, pp. 893–902.

[17] Jean-François Biasse, Thomas Espitau, Pierre-Alain Fouque, Alexandre Gélin, and Paul Kirchner. "Computing generator in cyclotomic integer rings - A subfield algorithm for the principal ideal problem in $\mathrm{L}_{|\Delta_{\mathbb{K}}|}(\frac{1}{2})$ and application to the cryptanalysis of a FHE scheme." In: *Advances in Cryptology – EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. Lecture Notes in Computer Science. Paris, France: Springer, Heidelberg, Germany, 2017, pp. 60–88.

[18] Hans Frederick Blichfeldt. "The minimum values of positive quadratic forms in six, seven and eight variables." In: *Mathematische Zeitschrift* 39.1 (1935), pp. 1–15.

[19] Leo Bluestein. "A linear filtering approach to the computation of discrete Fourier transform." In: *IEEE Transactions on Audio and Electroacoustics* 18.4 (1970), pp. 451–455.

[20] Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. "LWE without modular reduction and improved side-channel attacks against BLISS." In: *Advances in Cryptology – ASIACRYPT 2018, Part I*. Ed. by Thomas Peyrin and Steven Galbraith.

Vol. 11272. Lecture Notes in Computer Science. Brisbane, Queensland, Australia: Springer, Heidelberg, Germany, 2018, pp. 494–524.

[21] Thomas Borek. "Arakelov theory of noncommutative arithmetic curves." In: *Journal of Number Theory* 131.2 (2011), pp. 212–227.

[22] Jean-Benoît Bost. "Theta invariants of euclidean lattices and infinite-dimensional hermitian vector bundles over arithmetic curves." In: *arXiv preprint arXiv:1512.08946* (2015).

[23] N. Bourbaki. *Eléments de mathématique : Algèbre commutative: chapitres 1 à 4*. Eléments de mathématiques. Masson, 1985.

[24] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. "Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme." In: *Cryptographic Hardware and Embedded Systems – CHES 2016*. Ed. by Benedikt Gierlichs and Axel Y. Poschmann. Vol. 9813. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2016, pp. 323–345.

[25] Johannes A. Buchmann. "A subexponential algorithm for the determination of class groups and regulators of algebraic number fields." In: *Séminaire de théorie des nombres, Paris* 1989.1990 (1988), pp. 27–41.

[26] Johannes A. Buchmann. "Reducing lattice bases by means of approximations." In: *Algorithmic Number Theory, First International Symposium, ANTS-I, Ithaca, NY, USA, May 6-9, 1994, Proceedings*. 1994, pp. 160–168.

[27] Johannes A. Buchmann and Hendrik W Lenstra. "Approximatting rings of integers in number fields." In: *Journal de théorie des nombres de Bordeaux* 6.2 (1994), pp. 221–260.

[28] Johannes Buchmann. "A subexponential algorithm for the determination of class groups and regulators of algebraic number fields." In: *Séminaire de Théorie des Nombres, Paris 1988-1989* (1990), pp. 27–41.

[29] Peter Campbell, Michael Groves, and Dan Shepherd. *SOLILOQUY: A cautionary tale*. ETSI 2nd Quantum-Safe Crypto Workshop. http://docbox.etsi.org/workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves.pdf. 2014.

[30] Thomas Camus. "Méthodes algorithmiques pour les réseaux algébriques." In: (2017). Thèse de doctorat dirigée par Elbaz-Vincent, Philippe Mathématiques Grenoble Alpes 2017.

[31] E. J. Candés, X. Li, and M. Soltanolkotabi. "Phase Retrieval via Wirtinger Flow: Theory and Algorithms." In: *IEEE Transactions on Information Theory* 61.4 (2015), pp. 1985–2007.

[32] Earl R. Canfield, Paul Erdős, and Carl Pomerance. "On a problem of Oppenheim concerning 'factorisatio numerorum'." In: *Journal of Number Theory* 17 (1983), pp. 1–28.

[33] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: better lattice security estimates." In: *Advances in Cryptology – ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. Lecture Notes in Computer Science. Seoul, South Korea: Springer, Heidelberg, Germany, 2011, pp. 1–20.

[34] Henri Cohen. *A course in computational algebraic number theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1993. ISBN: 0-387-55640-0.

[35] Henri Cohen. *Advanced topics in computational number theory*. Vol. 193. Springer Science & Business Media, 2012.

[36] Henry Cohn and Abhinav Kumar. "Optimality and uniqueness of the Leech lattice among lattices." In: *Annals of Mathematics* (2009), pp. 1003–1050.

[37] John Horton Conway and Neil James Alexander Sloane. *Sphere packings, lattices and groups*. Vol. 290. Springer Science & Business Media, 2013.

[38] Don Coppersmith. "Finding a small root of a univariate modular equation." In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 1996, pp. 155–165.

[39] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. "Zeroizing attacks on indistinguishability obfuscation over CLT13." In: *PKC 2017: 20th International Conference on Theory and Practice of Public Key Cryptography, Part I*. Ed. by Serge Fehr. Vol. 10174. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Heidelberg, Germany, 2017, pp. 41–58.

[40] Renaud Coulangeon. "Voronoï theory over algebraic number fields." In: *Réseaux euclidiens, designs sphériques et formes modulaires* 37 (2001), pp. 147–162.

[41] Renaud Coulangeon and Takao Watanabe. "Hermite constant and Voronoi theory over a quaternion skew field." In: *Osaka Journal of Mathematics* 43.3 (2006), pp. 517–556.

[42] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. "Recovering short generators of principal ideals in cyclotomic rings." In: *Advances in Cryptology – EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. Lecture Notes in Computer Science. Vienna, Austria: Springer, Heidelberg, Germany, 2016, pp. 559–585.

[43] Richard Dedekind. "Uber den zusammenhang zwischen der theorie der ideale und der theorie der hoheren kongruenzen." In: *Abhandlungen der Koniglichen Gesellschaft der Wissenschaften zu Gottingen* 23 (1878), pp. 1–23.

[44] Whitfield Diffie and Martin Hellman. "New directions in cryptography." In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[45] Jintai Ding, Seungki Kim, Tsuyoshi Takagi, and Yuntao Wang. *LLL and stochastic sandpile models*. Cryptology ePrint Archive, Report 2019/1009. 2019.

[46] John D. Dixon. "Exact solution of linear quations using p-adic expansions." In: *Numerische Mathematik* 40 (1982), pp. 137–141.

[47] Léo Ducas and Tancrède Lepoint. *BLISS: Bimodal lattice signature schemes*. (proof-of-concept implementation). June 2013. URL: http://bliss.di.ens.fr/bliss-06-13-2013.zip.

[48] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. "Efficient identity-based encryption over NTRU lattices." In: *ASIACRYPT*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8874. LNCS. Springer, 2014, pp. 22–41.

[49] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. "Lattice signatures and bimodal gaussians." In: *CRYPTO*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, 2013, pp. 40–56.

[50] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. "Lattice signatures and bimodal gaussians." In: *Advances in Cryptology – CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2013, pp. 40–56.

[51] Peter van Emde Boas. "Another NP-complete partition problem and the complexity of computing short vectors in a lattice." In: *Technical Report* (Jan. 1981).

[52] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. "Loop-Abort Faults on Lattice-Based Signature Schemes and Key Exchange Protocols." In: *IEEE Trans. Computers* 67.11 (2018), pp. 1535–1549.

[53] Claus Fieker and Michael E. Pohst. "On lattices over number fields." In: *International Algorithmic Number Theory Symposium*. Springer. 1996, pp. 133–139.

[54] Claus Fieker and Damien Stehlé. "Short bases of lattices over number fields." In: *Algorithmic Number Theory, 9th International Symposium, ANTS-IX*. 2010, pp. 157–173.

[55] Nicolas Gama, Nick Howgrave-Graham, and Phong Q. Nguyen. "Symplectic lattice reduction and NTRU." In: *Advances in Cryptology – EUROCRYPT 2006*. Ed. by Serge Vaudenay. Vol. 4004. Lecture Notes in Computer Science. St. Petersburg, Russia: Springer, Heidelberg, Germany, 2006, pp. 233–253.

[56] Nicolas Gama and Phong Q. Nguyen. "Finding short lattice vectors within Mordell's inequality." In: *40th Annual ACM Symposium on Theory of Computing*. Ed. by Richard E. Ladner and Cynthia Dwork. Victoria, BC, Canada: ACM Press, 2008, pp. 207–216.

[57] Nicolas Gama and Phong Q. Nguyen. "Predicting lattice reduction." In: *Advances in Cryptology – EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. Lecture Notes in Computer Science. Istanbul, Turkey: Springer, Heidelberg, Germany, 2008, pp. 31–51.

[58] Sanjam Garg, Craig Gentry, and Shai Halevi. "Candidate multilinear maps from ideal lattices." In: *Advances in Cryptology - EUROCRYPT 2013, Proceedings*. 2013, pp. 1–17.

[59] Joachim von zur Gathen and Daniel Panario. "Factoring polynomials over finite fields: a survey." In: *Journal of Symbolic Computation* 31 (2001), pp. 3 –17. ISSN: 0747-7171.

[60] Carl Friedrich Gauss. *Disquisitiones arithmeticae*. Vol. 157. Yale University Press, 1966.

[61] Alexandre Gélin and Antoine Joux. "Reducing the complexity for class group computations using small defining polynomials." In: *To appear* (2018).

[62] Craig Gentry. "Fully homomorphic encryption using ideal lattices." In: *41st Annual ACM Symposium on Theory of Computing*. Ed. by Michael Mitzenmacher. Bethesda, MD, USA: ACM Press, 2009, pp. 169–178.

[63] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions." In: *STOC*. Ed. by Cynthia Dwork. ACM, 2008, pp. 197–206.

[64] Craig Gentry and Michael Szydlo. "Cryptanalysis of the revised NTRU signature scheme." In: *Advances in Cryptology – EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Vol. 2332. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Heidelberg, Germany, 2002, pp. 299–320.

[65] Craig Gentry and Michael Szydlo. "Cryptanalysis of the revised NTRU signature scheme." In: *EUROCRYPT*. Ed. by Lars R. Knudsen. Vol. 2332. LNCS. Springer, 2002, pp. 299–320.

[66] Craig Gentry, Jakob Jonsson, Jacques Stern, and Michael Szydlo. "Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001." In: *ASIACRYPT*. Ed. by Colin Boyd. Vol. 2248. LNCS. Springer, 2001, pp. 1–20.

[67] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. "Public-Key Cryptosystems from Lattice Reduction Problems." In: *CRYPTO*. Ed. by Burton S. Kaliski Jr. Vol. 1294. LNCS. Springer, 1997, pp. 112–131.

[68] Oded Goldreich and Leonid A Levin. "A hard-core predicate for all one-way functions." In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. ACM. 1989, pp. 25–32.

[69] Xavier Gourdon. "Combinatoire, algorithmique et géométrie des polynomes." In: *PhD thesis* (1996), pp. 27–49.

[70] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. "Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems." In: *CHES*. Ed. by Emmanuel Prouff and Patrick Schaumont. Vol. 7428. LNCS. Springer, 2012, pp. 530–547.

[71] James L. Hafner and Kevin S. McCurley. "A rigorous subexponential algorithm for computation of class groups." In: *Journal of American Mathematical Society* 2 (1989), pp. 839–850.

[72] Thomas C Hales. "A proof of the Kepler conjecture." In: *Annals of mathematics* (2005), pp. 1065–1185.

[73] Thomas C Hales. "Introduction to the Flyspeck project." In: *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2006.

[74] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. "Analyzing blockwise lattice algorithms using dynamical systems." In: *Advances in Cryptology – CRYPTO 2011*. Ed. by Phillip Rogaway. Vol. 6841. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2011, pp. 447–464.

[75] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. *Terminating BKZ*. Cryptology ePrint Archive, Report 2011/198. http://eprint.iacr.org/2011/198. 2011.

[76] Günter Harder and Mudumbai S Narasimhan. "On the cohomology groups of moduli spaces of vector bundles on curves." In: *Mathematische Annalen* 212.3 (1975), pp. 215–248.

[77] Christian Heckler and Lothar Thiele. "Complexity analysis of a parallel lattice basis reduction algorithm." In: *SIAM Journal on Computing* 27.5 (1998), pp. 1295–1302.

[78] C. Hermite. "Extraits de lettres de M. Ch. Hermite à M. Jacobi sur différents objects de la théorie des nombres. (Continuation)." fre. In: *Journal für die reine und angewandte Mathematik* 40 (1850), pp. 279–315.

[79] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. Vol. 80. Siam, 2002.

[80] Edmund Hlawka. "Zur geometrie der zahlen." In: *Mathematische Zeitschrift* 49.1 (1943), pp. 285–312.

[81] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem." In: *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*. 1998, pp. 267–288.

[82] Jeffrey Hoffstein, Jill Pipher, and Joseph Silverman. "NTRU: A ring-based public key cryptosystem." In: *ANTS*. 1998, pp. 267–288.

[83] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. "NTRUSign: digital signatures using the NTRU lattice." In: *CT-RSA*. Ed. by Marc Joye. Vol. 2612. LNCS. Springer, 2003, pp. 122–140.

[84] Jeffrey Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, and William Whyte. "Practical signatures from the partial fourier recovery problem." In: *ACNS*. Ed. by Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay. Vol. 8479. LNCS. Springer, 2014, pp. 476–493.

[85] James Howe, Thomas Pöppelmann, Máire O'Neill, Elizabeth O'Sullivan, and Tim Güneysu. "Practical Lattice-Based Digital Signature Schemes." In: *ACM Trans. Embedded Comput. Syst.* 14.3 (2015), p. 41.

[86] Nick Howgrave-Graham. "A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU." In: *Advances in Cryptology – CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2007, pp. 150–169.

[87] Nick Howgrave-Graham and Michael Szydlo. "A method to solve cyclotomic norm equations." In: *ANTS*. Ed. by Duncan A. Buell. Vol. 3076. LNCS. Springer, 2004, pp. 272–279.

[88] Pierre Humbert. "Théorie de la réduction des formes quadratiques définies positives dans un corps algébrique K fini." In: *Commentarii mathematici Helvetici* 12 (1939/40), pp. 263–306.

[89] Pierre Humbert. "Réduction de formes quadratiques dans un corps algébrique fini." In: *Commentarii Mathematici Helvetici* 23.1 (1949), pp. 50–63.

[90] Maria Ines Icaza. "Hermite constant and extreme forms for algebraic number fields." In: *Journal of the London Mathematical Society* 55.1 (1997), pp. 11–22.

[91] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Eric Walter. *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Springer Verlag, 2001.

[92] Jean-Pierre Kahane. "Local properties of functions in terms of random Fourier series." In: *Stud. Math* 19 (1960), pp. 1–25.

[93] Ravindran Kannan and Achim Bachem. "Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix." In: *siam Journal on Computing* 8.4 (1979), pp. 499–507.

[94] Subhash Khot. "Hardness of approximating the shortest vector problem in lattices." In: *Journal of the ACM (JACM)* 52.5 (2005), pp. 789–808.

[95] Seungki Kim and Akshay Venkatesh. *The behavior of random reduced bases*. 2016. eprint: arXiv:1608.00767.

[96] Paul Kirchner. *Algorithms on ideal over complex multiplication order.* Cryptology ePrint Archive, Report 2016/220. http://eprint.iacr.org/2016/220. 2016.

[97] Neal Koblitz. "Elliptic curve cryptosystems." In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[98] Keith Konrad. "A non-free relative integral extension." In: *Expository note* (2016). URL: https://kconrad.math.uconn.edu/blurbs/gradnumthy/notfree.pdf.

[99] A. Korkine and G. Zolotareff. "Sur les formes quadratiques." fre. In: *Mathematische Annalen* 6 (1873), pp. 366–389.

[100] Aleksandr Korkine and Yegor Zolotareff. "Sur les formes quadratiques positives quaternaires." In: *Mathematische Annalen* 5.4 (1872), pp. 581–583.

[101] Radan Kučera. "On bases of the Stickelberger ideal and of the group of circular units of a cyclotomic field." In: *Journal of Number Theory* 40.3 (1992), pp. 284–316.

[102] Joseph Louis Lagrange. "Recherches d'arithmétique." In: *Nouveaux mémoires de l'académie de Berlin* (1773).

[103] Edmund Landau. "Neuer beweis des primzahlsatzes und beweis des primidealsatzes." In: *Mathematische Annalen* 56 (1903), pp. 645–670.

[104] S. Lang and S.A. Lang. *Algebraic number theory.* Applied Mathematical Sciences. Springer, 1994.

[105] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices." In: *Designs, Codes and Cryptography* 75.3 (2015), pp. 565–599.

[106] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. "GGHLite: more efficient multilinear maps from ideal lattices." In: *Advances in Cryptology - EUROCRYPT 2014, Proceedings.* 2014, pp. 239–256.

[107] Hendrik W. Lenstra Jr. "Factoring integers with elliptic curves." In: *Annals of Mathematics* 126 (1987), pp. 649–673.

[108] Arjen K Lenstra and Hendrik W Lenstra Jr. "Algorithms in number theory." In: *Algorithms and Complexity.* Elsevier, 1990, pp. 673–715.

[109] Arjen K. Lenstra, Hendrik W. Jr. Lenstra, and László Lovász. "Factoring polynomials with rational coefficients." In: *Math. Ann.* 261 (1982), pp. 515–534.

[110] Arjen K. Lenstra, Hendrik W. Lenstra Jr., Mark S. Manasse, and John M. Pollard. "The number field sieve." In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing STOC 1990.* 1990, pp. 564–572.

[111] Hendrik W Lenstra. "Euclid's algorithm in cyclotomic fields." In: *Journal of the London Mathematical Society* 2.4 (1975), pp. 457–465.

[112] Vadim Lyubashevsky. "Fiat–Shamir with aborts: applications to lattice and factoring-based signatures." In: *ASIACRYPT*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, 2009, pp. 598–616.

[113] Vadim Lyubashevsky. "Lattice signatures without trapdoors." In: *EUROCRYPT*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, 2012, pp. 738–755.

[114] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On ideal lattices and learning with errors over rings." In: *Journal of the ACM* 60.6 (2013).

[115] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, and Damien Stehlé. *CRYSTALS-DILITHIUM*. Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions. National Institute of Standards and Technology, 2017.

[116] Manfred Madritsch and Brigitte Vallée. "Modelling the LLL algorithm by sandpiles." In: *Latin American Symposium on Theoretical Informatics*. Springer. 2010, pp. 267–281.

[117] David K Maslen and Daniel N Rockmore. "Generalized FFTs—a survey of some recent results." In: *Groups and Computation II*. Vol. 28. American Mathematical Soc. 1997, pp. 183–287.

[118] Robert J McEliece. "A public-key cryptosystem based on algebraic." In: *Coding Thv* 4244 (1978), pp. 114–116.

[119] Kurt Mehlhorn and Peter Sanders. *Algorithms and data structures: The basic toolbox*. Springer Science & Business Media, 2008.

[120] D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective*. The Springer International Series in Engineering and Computer Science. Springer US, 2002.

[121] Daniele Micciancio and Michael Walter. "Practical, predictable lattice basis reduction." In: *Advances in Cryptology – EUROCRYPT 2016, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9665. Lecture Notes in Computer Science. Vienna, Austria: Springer, Heidelberg, Germany, 2016, pp. 820–849.

[122] Hermann Minkowski. *Geometrie der zahlen*. Leipzig B.G. Teubner, 1910.

[123] Robert Moenck and Allan Borodin. "Fast modular transforms via division." In: *13th Annual Symposium on Switching and Automata Theory (swat 1972)*. IEEE. 1972, pp. 90–96.

[124] Ramon E. Moore. "Methods and applications of interval analysis." In: *Methods and Applications of Interval Analysis*. 1977.

[125] Ramon Moore. "Interval Arithmetic and automatic error analysis in digital computing." PhD thesis. Stanford, 1962.

[126] Huguette Napias. "A generalization of the LLL-algorithm over euclidean rings or orders." en. In: *Journal de théorie des nombres de Bordeaux* 2 (1996), pp. 387–396.

[127] Jürgen Neukirch. *Algebraic Number Theory*. Springer, 1988.

[128] Arnold Neumaier and Damien Stehlé. "Faster LLL-type reduction of lattice bases." In: *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC*. 2016, pp. 373–380.

[129] Phong Q. Nguyen and Oded Regev. "Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures." In: *J. Cryptology* 22.2 (2009), pp. 139–160.

[130] Phong Q. Nguyen and Damien Stehlé. "LLL on the average." In: *Algorithmic Number Theory, 7th International Symposium, ANTS-VII*. 2006, pp. 238–256.

[131] Phong Q Nguyen and Damien Stehlé. "An LLL algorithm with quadratic complexity." In: *SIAM Journal on Computing* 39.3 (2009), pp. 874–903.

[132] *PARI/GP, version 2.7.6*. http://pari.math.u-bordeaux.fr/. The PARI Group. Bordeaux, 2016.

[133] Chris Peikert. "A decade of lattice cryptography." In: *Foundations and Trends in Theoretical Computer Science* 10.4 (2016), pp. 283–424.

[134] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. "To BLISS-B or not to be: Attacking strongSwan's Implementation of Post-Quantum Signatures." In: *ACM CCS 2017: 24th Conference on Computer and Communications Security*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. Dallas, TX, USA: ACM Press, 2017, pp. 1843–1855.

[135] M Pohst. "A modification of the LLL reduction algorithm." In: *Journal of Symbolic Computation* 4.1 (1987), pp. 123–127.

[136] Carl Pomerance. "Analysis and comparison of some integer factoring algorithms." In: *Computational Methods in Number Theory* (1982), pp. 89–139.

[137] Thomas Pöppelmann, Tobias Oder, and Tim Güneysu. "High-Performance Ideal Lattice-Based Cryptography on 8-Bit ATxmega Microcontrollers." In: *LATINCRYPT*. Ed. by Kristin E. Lauter and Francisco Rodríguez-Henríquez. Vol. 9230. LNCS. Springer, 2015, pp. 346–365.

[138] Helmut Ratschek and Jon Rokne. *New computer methods for global optimization*. New York, NY, USA: Halsted Press, 1992.

[139] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography." In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing STOC 2005*. 2005, pp. 84–93.

[140] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography." In: *Journal of the ACM (JACM)* 56.6 (2009), p. 34.

[141] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." In: *Commununications of the ACM* 21.2 (1978), pp. 120–126.

[142] P. Samuel. *Théorie algébrique des nombres*. Collection Méthodes. Hermann, 1967.

[143] P. Sawyer. "Computing the Iwasawa decomposition of the classical Lie groups of noncompact type using the QR decomposition." In: *Linear Algebra and its Applications* 493 (2016), pp. 573–579.

[144] John Schanck. *LOGCVP, Pari implementation of CVP in Log embedding*. https://github.com/jschanck-si/logcvp. 2015.

[145] Claus-Peter Schnorr. "A hierarchy of polynomial time lattice basis reduction algorithms." In: *Theoretical computer science* 53.2-3 (1987), pp. 201–224.

[146] Claus-Peter Schnorr. "A more efficient algorithm for lattice basis reduction." In: *J. Algorithms* 9.1 (1988), pp. 47–62.

[147] Claus-Peter Schnorr and Michael. Euchner. "Lattice basis reduction: Improved practical algorithms and solving subset sum problems." In: *Math. Program.* 66 (1994), pp. 181–199.

[148] Arnold Schönhage. "Fast reduction and composition of binary quadratic forms." In: *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*. ISSAC '91. Bonn, West Germany, 1991, pp. 128–133.

[149] René Schoof. "Computing Arakelov class groups." In: *arXiv preprint arXiv:0801.3835* (2008).

[150] Martin Seysen. "A probabilistic factorization algorithm with quadratic forms of negative discriminant." In: *Mathematics of Computation* 84 (1987), pp. 757–780.

[151] Daniel Shanks. "The infrastructure of a real quadratic field and its applications." In: *Proc. 1972 Number Theory Conf., Boulder, Colorado*. 1972, pp. 217–224.

[152] Denis Simon. "Solving norm equations in relative number fields using S-units." In: *Mathematics of computation* 71.239 (2002), pp. 1287–1305.

[153] Nigel P. Smart and Frederik Vercauteren. "Fully homomorphic encryption with relatively small key and ciphertext sizes." In: *Public Key Cryptography - PKC 2010, Proceedings*. 2010, pp. 420–443.

[154] John Stillwell. *Classical topology and combinatorial group theory*. Vol. 72. Springer Science & Business Media, 2012.

[155] Arne Storjohann. "The shifted number system for fast linear algebra on integer matrices." In: *Journal of Complexity* 21.4 (2005), pp. 609–650.

[156] Arne Storjohann and George Labahn. "Asymptotically fast computation of Hermite normal forms of integer matrices." In: *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*. ACM. 1996, pp. 259–266.

[157] Ji-Guang Sun. "Perturbation bounds for the Cholesky and QR factorizations." In: *BIT Numerical Mathematics* 31.2 (1991), pp. 341–352.

[158] Teruo Sunaga. "Theory of an interval algebra and its application to numerical analysis." In: *Japan J. Indust. Appl. Math.* 26 (Oct. 2009).

[159] The FPLLL Development Team. "FPLLL, a lattice reduction library." Available at https://github.com/fplll/fplll. 2016. URL: https://github.com/fplll/fplll.

[160] Christoph Thiel. "On the complexity of some problems in algorithmic algebraic number theory." https://www.cdc.informatik.tu-darmstadt.de/reports/reports/Christoph_Thiel.diss.pdf. PhD thesis. Universität des Saarlandes, 1995.

[161] Gilles Villard. "Parallel lattice basis reduction." In: *Proceedings of the 1992 International Symposium on Symbolic and Algebraic Computation, ISSAC '92, Berkeley, CA, USA, July 27-29, 1992*. 1992, pp. 269–277.

[162] Gilles Villard. *Certification of the QR factor R, and of lattice basis reducedness*. 2007. eprint: arXiv:cs/0701183.

[163] Lawrence C. Washington. *Introduction to cyclotomic fields*. Germany: Springer, 1997.

[164] B. M. M. De Weger. "Solving exponential Diophantine equations using lattice basis reduction algorithms." In: *J. Number theory* 26 (1987), pp. 325–367.

[165] Rosalind Cecil Young. "The algebra of many-values quantities." PhD thesis. Cambridge, 1931.

Part IV

APPENDIX

1

OMITTED PROOFS OF CHAPTER 3

This first appendix provides the proof of the technical lemmas related to the properties of covolumes and degrees.

1.1 PROOF OF THE COMPATIBILITY OF DEGREE WITH DIRECT SUM AND TENSOR PRODUCT

Lemma 1.1.1. *2.1.4 Let Λ, Λ' two lattices. Then:*

1. $\deg(\Lambda \otimes_{\mathbf{Z}} \Lambda') = \operatorname{rk}\Lambda' \deg\Lambda + \operatorname{rk}\Lambda' \deg\Lambda'$
2. $\deg(\Lambda \oplus \Lambda') = \deg\Lambda + \deg\Lambda'$.

Proof. Let Λ, Λ' two lattices of rank respectively d, d'. Recall that the exterior algebra $\bigwedge\Lambda$ over a $\mathbf{Z}$-module Λ is defined as the quotient algebra of the tensor algebra $T(\Lambda)$ by the two-sided ideal I generated by all elements of the form $v \otimes v$ for $v \in \Lambda$ (i.e. all tensors that can be expressed as the tensor product of any vector in Λ by itself). The k exterior power of Λ, denoted $\bigwedge^k \Lambda$, is the vector subspace of $\bigwedge\Lambda$ spanned by elements of the form $v_1 \wedge v_2 \wedge \cdots \wedge v_k$, where $v_i \in \Lambda$ for $i = 1, 2, \ldots, k$. Canonically, the $i-th$ exterior product can be endowed with a Euclidean norm, defined as

$$\|v_1 \wedge \cdots \wedge v_k\| = \det(\langle v_i, v_j \rangle)_{1 \leq i \leq j \leq k}.$$

1. Remark that:

$$\bigwedge^{dd'}(\Lambda \otimes_{\mathbf{Z}} \Lambda') \cong \left(\bigwedge^{d}\Lambda\right)^{\otimes d'} \otimes \left(\bigwedge^{d'}\Lambda'\right)^{\otimes d},$$

   so that we can reduce to prove the announced relation for lattices of rank 1. Suppose then that $dd' = 1$. By definition of the metric on the tensor product for any vectors $(v, v') \in \Lambda \times \Lambda'$ we have $\|v \otimes v'\| = \|v\|\|v'\|$. Taking the logarithm then conclude since $\Lambda \times \Lambda'$ is of rank 1.

2. We have by [23], A.III §7.10:

$$\bigwedge^{d+d'}(\Lambda \oplus \Lambda') \cong \left(\bigwedge^{d}\Lambda\right) \otimes \left(\bigwedge^{d'}\Lambda'\right).$$

   It then suffices to use the point (1) we just proved.

■

1.2 PROOF OF THE COMPATIBILITY OF DEGREE WITH DUALITY

Lemma 1.2.1. *2.1.5 Let Λ a lattice and Λ' a sublattice of Λ, then:* $\deg \Lambda = -\deg \Lambda^\vee$.

To complete this proof we first need a technical lemma, which is a consequence of the compatibility of the degree with tensor products.

Corollary 1.2.1. *Let Λ a lattice and Λ' a sublattice of Λ. The degree is additive over the short exact sequence:*

$$0 \longrightarrow \Lambda' \longrightarrow \Lambda \longrightarrow {}^{\Lambda}\!/_{\Lambda'} \longrightarrow 0,$$

that is:

$$\deg \Lambda = \deg \Lambda' + \deg {}^{\Lambda}\!/_{\Lambda'}$$

Proof. By the result (2) of Lemma 2.1.4 it suffices to prove that the sequence splits, that is: $\Lambda' \oplus {}^{\Lambda}\!/_{\Lambda'} \cong \Lambda$. ∎

Proof. Proof of Lemma 2.1.5 Let Λ a lattice and $\Lambda^\vee$ its dual, then $\Lambda \otimes \Lambda^\vee \cong \mathbf{Z}$. The result follows from (2) of Lemma 2.1.4 since $\deg \mathbf{Z} = 0$. ∎

2

ON THE REDUCED BASES OF THE Λ_3 LATTICE

Let $\alpha \in]1, \sqrt{4/3}]$ and define $b_1 = (\alpha^2 \quad 0 \quad 0), b_2 = (\alpha\sqrt{\alpha^2-1} \quad \alpha \quad 0)$ and $b_3 = (0 \quad \sqrt{\alpha^2-1} \quad 1)$. We denote by Λ_3 the Euclidean lattice spanned by b_1, b_2, b_3. The aim of this appendix is to prove that this lattice has only two LLL-reduced bases : $L_3 = [b_1, b_2, b_3]$ and the size-reduction of $R_3 = [b_3, b_2, b_1]$.

We start by enumerating the short vectors of Λ_3 and its dual.

2.1 ON THE GEOMETRY OF SMALL VECTORS IN Λ_3 AND $\Lambda_3^\vee$

2.1.1 Short vectors of Λ_3

Lemma 2.1.1. $B\left(0, \alpha\sqrt{\frac{4}{3}}\right) \cap \Lambda_3 = \{0, \pm b_1, \pm b_2, \pm b_3\}$.

Claim 1. *The Gram-Schmidt orthogonalization of $R_3 = [b_3, b_2, b_1]$ is*

$$\begin{bmatrix} 0 & \sqrt{\alpha^2-1} & 1 \\ \alpha\sqrt{\alpha^2-1} & \frac{1}{\alpha} & -\frac{\sqrt{\alpha^2-1}}{\alpha} \\ \frac{\alpha^2}{\alpha^4-\alpha^2+1} & -\frac{\alpha^2\sqrt{\alpha^2-1}}{\alpha^4-\alpha^2+1} & \frac{(\alpha^2-1)\alpha^2}{\alpha^4-\alpha^2+1} \end{bmatrix} \begin{pmatrix} = & b_3 & := r_3^* \\ = & \pi_{(b_3)^\perp}(b_2) & := r_2^* \\ = & \pi_{(b_3,b_2)^\perp}(b_1) & := r_1^* \end{pmatrix}$$

and we have:

$$\frac{\langle b_2, r_3^*\rangle}{\|r_3^*\|^2} = \frac{\sqrt{\alpha^2-1}}{\alpha} \qquad \frac{\langle b_1, r_2^*\rangle}{\|r_2^*\|^2} = \frac{\sqrt{\alpha^2-1}\alpha^3}{\alpha^4-\alpha^2+1} \qquad \frac{\langle b_3, r_1^*\rangle}{\|r_1^*\|^2} = 0$$

Proof. Direct computation from the matrix of the basis R_3. ∎

Proof of Lemma 2.1.1. Let $v = x_1 \cdot b_1 + x_2 \cdot b_2 + x_3 \cdot b_3$ be a vector of the ball of radius $\sqrt{\frac{4\alpha^2}{3}}$, with $x_1, x_2, x_3 \in \mathbf{Z}$. By Gram-Schmidt orthogonalization we have: $v = z_1 \cdot r_1^* + z_2 \cdot r_2^* + z_3 \cdot r_3^*$ where $z_1 = x_1, z_2 = x_2 + \frac{\langle b_1, r_2^*\rangle}{\|r_2^*\|^2} x_1$ and $z_3 = x_3 + \frac{\langle b_2, r_3^*\rangle}{\|r_3^*\|^2} x_2$. By orthogonality: $z_1^2\|r_1^*\|^2 + z_2^2\|r_2^*\|^2 + z_3^2\|r_3^*\|^2 \leq \frac{4\alpha^2}{3}$, so that

$$x_1^2 \frac{\alpha^4}{\alpha^4-\alpha^2+1} = z_1^2\|r_1^*\|^2 \leq \frac{4\alpha^2}{3}.$$

Hence $z_1^2 \leq \frac{13}{9}$ and all in all, $x_1 \in \{-1, 0, 1\}$.

Case $x_1 = 0$: We have $z_2^2\|r_2^*\|^2 + z_3^2\|r_3^*\|^2 \leq \frac{4\alpha^2}{3}$ and as such

$$x_2^2 = z_2^2 \leq \frac{4}{3(\alpha^4-\alpha^2+1)} < 2$$

and so $x_2 \in \{-1, 0, 1\}$.

Case $x_2 = 0$: We have $z_3^2\|r_3^*\|^2 \leq \frac{4\alpha^2}{3}$ and as such

$$x_3^2\alpha^2 = z_3^2\|r_3^*\|^2 \leq \frac{4\alpha^2}{3}$$

and so $x_3 \in \{-1, 0, 1\}$, leading to $v \in \{0, \pm b_3\}$.

Case $x_2^2 = 1$: In this case we have:

$$\left(x_3 - \frac{\sqrt{\alpha^2-1}}{\alpha}\right)^2 \alpha^2 = z_3^2\alpha^2 \leq \frac{4\alpha^2}{3} - (\alpha^4 - \alpha^2 + 1) \leq \frac{7\alpha^2}{12}$$

leading to:

$$-2 < -\sqrt{\frac{7}{12}} + \frac{\sqrt{\alpha^2-1}}{\alpha} \leq x_3 \leq \sqrt{\frac{7}{12}} + \frac{\sqrt{\alpha^2-1}}{\alpha} < 2.$$

Therefore, up to sign, $v \in \{b_2, b_2 + b_3, b_2 - b_3\}$. Remark that:

$$\begin{aligned} \|b_2 + b_3\|^2 &= \left(\alpha^3 + \alpha + 2\sqrt{\alpha^2-1}\right)\alpha > \tfrac{4}{3}\alpha \\ \|b_2 - b_3\|^2 &= \left(\alpha^3 + \alpha - 2\sqrt{\alpha^2-1}\right)\alpha > \tfrac{4}{3}\alpha' \end{aligned}$$

since $\alpha > 1$. Hence $v = \pm b_2$.

Case $x_1^2 = 1$: Without loss of generality (since the lattice is invariant by sign change) we can suppose that $x_1 = 1$. We have $z_2^2\|r_2^*\|^2 + z_3^2\|r_3^*\|^2 \leq \frac{4\alpha^2}{3} - \frac{\alpha^4}{\alpha^4-\alpha^2+1} < \frac{7}{9}$ and as such

$$\left(x_2 - \frac{\sqrt{\alpha^2-1}\alpha^3}{\alpha^4 - \alpha^2 + 1}\right)^2 (\alpha^4 - \alpha^2 + 1) = z_2^2\|r_2^*\|^2 \leq \frac{7}{9}$$

and so: $-1 < x_2 < 1$. Therefore $x_2 = 0$. We have $z_3^2\|r_3^*\|^2 \leq \frac{12\alpha^2}{14}$ and as such

$$x_3^2\alpha^2 = z_3^2\|r_3^*\|^2 \leq \frac{12\alpha^2}{13}$$

and so $x_3 = 0$, leading to $v = b_1$.

The only vectors in the ball of desired radius are therefore the vectors b_1, b_2, b_3. ∎

Corollary 2.1.1. *The shortest vector of Λ_3 is unique up to sign: it is $\pm b_3$.*

Proof. b_3 is the shortest of b_1, b_2, b_3 which are the only vectors of norm smaller than $\sqrt{4/3}\alpha$. ∎

2.1.2 Short vectors of $\Lambda_3^\vee$

Lemma 2.1.2. $B\left(0, \alpha^{-1}\sqrt{\frac{4}{3}}\right) \cap \Lambda_3^\vee = \{0, \pm d_1, \pm d_2, \pm d_3\}$

In order to prove Lemma 2.1.2, we first compute a basis of the dual lattice from the basis L_3 and compute its Gram-Schmidt orthogonalization.

Claim 2. *Let $D = [d_3, d_2, d_1]$ the reversed dual base of L_3. Then:*

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & \frac{1}{a} & -\frac{\sqrt{a^2-1}}{a} \\ \frac{1}{a^2} & -\frac{\sqrt{a^2-1}}{a^2} & \frac{a^2-1}{a^2} \end{pmatrix}$$

Its Gram-Schmidt orthogonalization is:

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & \frac{1}{a} & 0 \\ \frac{1}{a^2} & 0 & 0 \end{pmatrix} \begin{pmatrix} = & d_3 & := d_3^* \\ = & \pi_{(d_3)^\perp}(d_2) & := d_2^* \\ = & \pi_{(d_3,d_2)^\perp}(d_1) & := d_1^* \end{pmatrix}$$

Then we have:

$$\frac{\langle d_2, d_3^* \rangle}{\|d_3^*\|^2} = -\frac{\sqrt{\alpha^2-1}}{\alpha} \qquad \frac{\langle d_1, d_2^* \rangle}{\|d_2^*\|^2} = -\frac{\sqrt{\alpha^2-1}}{\alpha} \qquad \frac{\langle d_3, d_1^* \rangle}{\|d_1^*\|^2} = 0$$

Proof of Lemma 2.1.2. Let $v = x_1 \cdot d_1 + x_2 \cdot b_2 + x_3 \cdot v_3$ be a vector of the ball of radius $\alpha^{-1}\sqrt{\frac{4}{3}}$ in $\Lambda^\vee$. By definition of the Gram-Schmidt orthogonalization we have $v = z_1 \cdot d_1^* + z_2 \cdot d_2^* + z_3 \cdot d_3^*$ where $z_1 = x_1, z_2 = x_2 + \frac{\langle d_1, d_2^* \rangle}{\|d_2^*\|^2} x_1$ and $z_3 = x_3 + \frac{\langle d_2, d_3^* \rangle}{\|d_3^*\|^2} x_2$. By orthogonality: $z_1^2\|d_1^*\|^2 + z_2^2\|d_2^*\|^2 + z_3^2\|d_3^*\|^2 \leq \frac{4}{3\alpha^2}$, so that

$$x_1^2 \frac{1}{\alpha^4} = z_1^2\|d_1^*\|^2 \leq \frac{4}{3\alpha^2},$$

meaning that $-\frac{4}{3} \leq x_1 \leq \frac{4}{3}$ and so that $x_1 \in \{-1, 0, 1\}$.

Case $x_1 = 0$: We have $z_2^2\|d_2^*\|^2 + z_3^2\|d_3^*\|^2 \leq \frac{4}{3\alpha^2}$ and as such

$$x_2^2 = z_2^2\|r_2^*\|^2 \leq \frac{4}{3\alpha^2}$$

and so $-\frac{4}{3} \leq x_2 \leq \frac{4}{3}$, meaning that $x_2 \in \{-1, 0, 1\}$.

Case $x_2 = 0$: We have $z_3^2\|d_3^*\|^2 \leq \frac{4}{3\alpha^2}$ and as such

$$x_3^2 = z_3^2\|r_3^*\|^2 \leq \frac{4\alpha^2}{3}$$

and so $x_3 \in \{-1, 0, 1\}$, leading to $v \in \{0, \pm d_3\}$.

Case $x_2^2 = 1$: In this case we have:

$$\left(x_3 + \frac{\sqrt{\alpha^2-1}}{\alpha}\right)^2 = z_3^2\|r_3^*\|^2 \leq \frac{4}{3\alpha^2} - \frac{1}{\alpha^2} \leq \frac{1}{3\alpha^2}$$

leading to:

$$-2 < -\sqrt{\frac{1}{3}} - \frac{\sqrt{\alpha^2-1}}{\alpha} \leq x_3 \leq \sqrt{\frac{1}{3}} - \frac{\sqrt{\alpha^2-1}}{\alpha} < 1.$$

Therefore, up to sign, $v \in \{d_2, d_2 - d_3\}$. Remark that:

$$\|r_2 - r_3\|^2 = \frac{2\sqrt{\alpha^2-1}}{\alpha} + 2 > \frac{4}{3\alpha^2}$$

since $\alpha > 1$. Hence $v = \pm d_2$.

Case $x_1^2 = 1$: Without loss of generality (since the lattice is invariant by sign change) we can suppose that $x_1 = 1$. We have $z_2^2\|r_2^*\|^2 + z_3^2\|r_3^*\|^2 \leq \frac{4}{3\alpha^2} + \frac{1}{\alpha^2} - \frac{1}{\alpha^4} - 1 = \frac{-3a^4+7a^2-3}{3a^4}$ and as such

$$\left(x_2 + \frac{\sqrt{\alpha^2-1}}{\alpha}\right)^2 \frac{1}{\alpha^2} = z_2^2\|r_2^*\|^2 \leq \frac{-3a^4+7a^2-3}{3a^4}$$

and so:

$$\left(x_2 + \frac{\sqrt{\alpha^2-1}}{\alpha}\right)^2 \leq \frac{10}{9}$$

$-2 < x_2 < 2$. Therefore $x_2 \in \{-1, 0, 1\}$.

Case $x_2 = 0$: We have $z_3^2\|d_3^*\|^2 \leq \frac{4}{3\alpha^2} - \frac{1}{\alpha^4}$ and as such

$$x_3^2 = z_3^2\|r_3^*\|^2 \leq \frac{4\alpha^2-1}{3\alpha^4},$$

and so $-2 < x_3 < 2$, that is $x_3 \in \{-1, 0, 1\}$. Remark now that:

- $\|d_1 + d_3\|^2 = \left(\frac{a^2-1}{a} + 1\right)^2 + \frac{a^2-1}{a^2} + \frac{1}{a^4} > \frac{4}{3a^2}$
- $\|d_1 - d_3\|^2 = \left(\frac{a^2-1}{a} - 1\right)^2 + \frac{a^2-1}{a^2} + \frac{1}{a^4} > \frac{4}{3a^2}$, since $\alpha > 0$

All in all $v = d_1$.

Case $x_2 = 1$: We have $z_3^2\|d_3^*\|^2 \leq \frac{4}{3\alpha^2} - \frac{1}{\alpha^4} - \frac{1}{a^2} < 0$, which is not possible.

Wrapping up everything gives $v \in \{\pm d_1, \pm d_2, \pm d_3\}$.

■

2.2 ON REDUCED BASIS OF THE LATTICE Λ_3

We are now able to prove our main theorem on the structure of reduced bases of Λ_3.

Theorem 2.2.1. *Let $\alpha \in]1, \sqrt{4/3}]$ and define $b_1 = (\alpha^2 \quad 0 \quad 0)$, $b_2 = (\alpha\sqrt{\alpha^2-1} \quad \alpha \quad 0)$ and $b_3 = (0 \quad \sqrt{\alpha^2-1} \quad 1)$, so that Λ_3 is spanned by these latter vectors. The respective (squared) norms of these vectors is $\alpha^4, \alpha^2, \alpha^4$. Then we have*

1. *The basis $L_3 = [b_1, b_2, b_3]$ is LLL reduced.*
2. *The basis $R_3 = [b_3, b_2, b_1]$ is weakly-HKZ reduced.*
3. *The only LLL reduced bases of the lattice Λ_3 are L_3 and $SR(R_3)$.*

Proof.

Claim 3. *Let $W = [\pi_{(b_3^\perp)}(b_2), \pi_{(b_3^\perp)}(b_1)]$ of $\pi_{(b_3^\perp)}(\Lambda_3)$. Then $SR(W)$ is Gauss-reduced.*

Proof. Recall that $\pi_{(b_3^\perp)}(b_2) = r_2^* = [a\sqrt{\alpha^2-1} \quad a - \frac{\alpha^2-1}{\alpha} \quad -\frac{\sqrt{\alpha^2-1}}{\alpha}]$. Moreover, $\pi_{(b_3^\perp)}(b_1) = b_1$ since b_1 and b_3 are orthogonals. Hence the Gram matrix of the basis $[r_2^*, b_1]$ is:

$$\begin{pmatrix} \alpha^4 - \alpha^2 + 1 & \sqrt{\alpha^2-1}\alpha^3 \\ \sqrt{\alpha^2-1}\alpha^3 & \alpha^4 \end{pmatrix}.$$

Clearly $\alpha^4 - \alpha^2 + 1 \leq \alpha^4$, so that $\|r_2^*\| < \|b_1\|$. Let us study the function $f : a \mapsto \frac{\sqrt{\alpha^2-1}\alpha^3}{\alpha^4-\alpha^2+1}$, which is the value of $\langle b_1, r_2^*\rangle / \langle r_2^*, r_2^*\rangle$: its derivative is $-\frac{(a^4-5a^2+3)a^2}{(a^4-\alpha^2+1)^2\sqrt{\alpha^2-1}}$ so that we have:

a	1		$\sqrt{\frac{5}{2}+\sqrt{\frac{13}{4}}}$		$+\infty$
$f'(a)$		$+$	0	$-$	
$f(a)$	0	$\nearrow$	$1 < \sqrt{\frac{35}{72}+\frac{13\sqrt{13}}{72}} < 2$	$\searrow$	1

These variations assert that $f - \frac{1}{2}$ has a unique real zero c_3 in the interval $[1, \sqrt{4/3}]$.

Case $\alpha \leq c_3$: Then $f(\alpha) \leq 1/2$, showing that the basis is indeed Gauss reduced since then $\langle b_1, r_2^*\rangle / \langle r_2^*, r_2^*\rangle < 1/2$.

Case $\alpha \geq c_3$: Then by monotonicity $8/13 = f(\sqrt{4/3}) > f(\alpha)$, meaning that $SR([r_2^*, b_1]) = [r_2^*, b_1 - r_2^*]$. But remark that:

$$\|b_1 - r_2^*\|^2 = 2\alpha^4 - 2\sqrt{\alpha^2-1}\alpha^3 - \alpha^2 + 1 > \alpha^4 - \alpha^2 + 1$$

since $\alpha \geq c_3$. Then $SR([r_2^*, b_1])$ is Gauss-reduced, which concludes the proof.

■

Corollary 2.2.1.

The basis $R_3 = [b_3, b_2, b_1]$ is weakly-HKZ reduced.

Proof. b_3 is the shortest vector of the lattice Λ_3 by Corollary 2.1.1 and $\pi_{(b_3)^\perp}(b_2)$ is the shortest vector of $\pi_{(b_3)^\perp}(\Lambda_3)$ as being the first vector of a Gauss-reduced basis, by Claim 3. ■

Corollary 2.2.2.

The basis $SR(R_3)$ is the only LLL-basis of Λ_3 of first vector b_3.

Proof. Since $\alpha > 1$, the study of the norm of the projected vectors $\pi_{(b_3)^\perp}(b_2)$ and $\pi_{(b_3)^\perp}(b_1)$ done in Claim 3 asserts that $\|\pi_{(b_3)^\perp}(b_2)\| < \|\pi_{(b_3)^\perp}(b_1)\|$. Since the vectors of a Gauss reduced basis reach respectively the first and second minimum of their rank 2 lattice we deduce that $\lambda_1(\pi_{(b_3)^\perp}(\Lambda_3)) < \lambda_2(\pi_{(b_3)^\perp}(\Lambda_3)$. Let $[b_3, x, y]$ a LLL reduced basis of Λ_3, then by definition the projected vectors $[\pi_{(b_3)^\perp}(x), \pi_{(b_3)^\perp}(y)]$ are a Gauss reduced basis of $\pi_{(b_3)^\perp}(\Lambda_3)$. Then $\pi_{(b_3)^\perp}(x) = r_2^*$ and $\pi_{(b_3)^\perp}(y) = b_1$. This implies that there exists two integers u, v such that $x = b_2 + ub_3$ and $y = b_1 + vb_2$. By size-reduceness of $[b_3, x, y]$, $u, v = 0$, which concludes the proof. ∎

Claim 4. *The projected basis $[\pi_{(b_1^\perp)}(b_2), \pi_{(b_1^\perp)}(b_3)]$ is Gauss-reduced in $\pi_{(b_1^\perp)}(\Lambda_3)$.*

Proof. We have: $x := \pi_{(b_1^\perp)}(b_2) = [0 \quad \alpha \quad 0]$ and $y := \pi_{(b_1^\perp)}(b_3) = b_3$ by orthogonality. Hence the Gram matrix of the projected basis $[x \quad y]$ is:

$$\begin{pmatrix} \alpha^2 & \sqrt{\alpha^2 - 1}\alpha \\ \sqrt{\alpha^2 - 1}\alpha & \alpha^2 \end{pmatrix}.$$

Therefore

$$\frac{\langle x, y \rangle}{\|x\|^2} = \frac{\sqrt{\alpha^2 - 1}}{\alpha} \leq \frac{1}{2}.$$

∎

Corollary 2.2.3.

The basis L_3 is LLL reduced.

Proof. By Claim 4 we only need to verify that $[b_1, b_2]$ is Gauss-reduced to conclude, which is immediate. ∎

Claim 5. *The shortest vectors of $\pi_{(b_1)^\perp}(\Lambda_3)$ are exactly $\pi_{(b_1)^\perp}(b_2), \pi_{(b_1)^\perp}(b_3)$.*

Proof. By Claim 4, the first two minima in this lattice coincides and are equals to α. Therefore if another shortest vector exists it is of the form $\pi_{(b_1)^\perp}(b_2) \pm \pi_{(b_1)^\perp}(b_3)$, which have squared norm respectively equal to: $2\alpha(\alpha \pm \sqrt{\alpha^2 - 1}) > \alpha^2$. ∎

Claim 6.

The basis L_3 is the only LLL-basis of Λ_3 of first vector b_1.

Proof. Let $[b_1, x, y]$ a LLL reduced basis of Λ_3, then by definition the projected vectors $[\pi_{(b_1)^\perp}(x), \pi_{(b_1)^\perp}(y)]$ are a Gauss reduced basis of $\pi_{(b_1)^\perp}(\Lambda_3)$, hence by size reduceness and Claim 6, either $x = b_2, y = b_3$ or $x = b_3, y = b_2$. But in the latter case, the Lovasz condition fails between b_3 and b_1 since $\|b_3\| < \|b_1\|$. ∎

Claim 7. *The size reduced projected basis $SR([\pi_{(b_2^\perp)}(b_1), \pi_{(b_2^\perp)}(b_1)])$ is Gauss-reduced in $\pi_{(b_2^\perp)}(\Lambda_3)$.*

Proof. A simple computation ensures that $\pi_{(b_2^\perp)}(b_3) = [-\frac{\alpha^2-1}{\alpha^2} \quad \frac{(\alpha^2-1)^{\frac{3}{2}}}{\alpha^2} \quad 1] := x$ and $\pi_{(b_3^\perp)}(b_1) = [1 \quad -\sqrt{\alpha^2-1} \quad 0] =: y$. Hence the Gram matrix of the projected basis $[x, y]$ is:

$$\begin{pmatrix} \alpha^2 + 1/\alpha^2 - 1 & 1-\alpha^2 \\ 1-\alpha^2 & \alpha^2 \end{pmatrix}.$$

Once again we introduce the function $f : a \mapsto \frac{a^2-1}{a^2+1/a^2-1}$. This function admits $\frac{4a^3-2a}{(1-a^2+a^4)^2}$ for derivative so that f is increasing between 1 and $+\infty$. Since $f(1) = 0$ and $\lim_\infty f(x) = 1$, we can ensure that for any $1 < \alpha < \sqrt{4/3}$ either $[x, y]$ or $[x, x+y]$ is size-reduced. The reasoning is then exactly the same as in Claim 3. ∎

Corollary 2.2.4.

There exists no LLL*-basis of* Λ_3 *of first vector* b_2.

Proof. Let $[b_2, x, y]$ be a LLL reduced basis of Λ_3 then $[\pi_{(b_2^\perp)}(x), \pi_{(b_2^\perp)}(y)]$ is Gauss-reduced. But this projected lattice is non critical by Claim 7 and as such we have: $\pi_{(b_2^\perp)}(x) = \pi_{(b_2^\perp)}(b_3)$ and $\pi_{(b_2^\perp)}(y) = \pi_{(b_2^\perp)}(b_1)$. Thus, by size-reduceness: $x = b_3, y = b_1$. But remark now that $\|b_2\| > \|b_3\|$, which contradicts the Lovász condition for these two vectors. Therefore there can not be a reduced basis starting with b_2. ∎

A LLL reduced basis of Λ_3 has its first vector contained in a ball of radius $\left(\frac{4}{3}\right)^{\frac{3-1}{4}} (\det \Lambda_3)^{\frac{1}{3}} = \frac{4\alpha}{3}$, so is one of the b_1, b_2, b_3 by Lemma 2.1.1. Then by Claim 6, Corollary 2.2.2,Corollary 2.2.4 the only bases starting with one of these vectors are L_3 and R_3. ∎

3

PRECISION REQUIRED TO REDUCE ALGEBRAIC LATTICES

In this appendix, we give details on the precision required the algorithms of Chapter 5. We first indicate the loss of precision of elementary operations, then look at the precision and complexity of the QR decomposition, and finally the size-reduction procedure. The penultimate part indicates how to use fast matrix multiplication to reach the same goal. We recall that w is the number of bits in the words. Eventually we adapt the proof of the reduction algorithm for cyclotomic fields to the symplectic context.

In this section, we give details on the precision required in our algorithms. We first indicate the loss of precision of elementary operations, then look at the precision and complexity of the QR decomposition, and finally the size-reduction procedure. The last part indicates how to use fast matrix multiplication to reach the same goal. We recall that w is the number of bits in the words.

3.1 ELEMENTARY OPERATIONS

3.1.1 Fast computation of primitive roots of unity

The fast Fourier transform algorithm needs a precise approximation of the primitive roots of unity to be performed in fixed-point arithmetic. In order to compute with high precision a primitive f-th root of unity, one can use Newton's method where we start with $1 + 6.3i/f$. The following lemma ensures that the convergence, in this case, is at least quadratic.

Lemma 3.1.1. *Let $x \in \mathbf{C}$ such that $|x| \geq 1 - \frac{1}{2f}$, then by setting $x' = x - \frac{x^f - 1}{f x^{f-1}}$ and with $\zeta^f = 1$, we have:*

$$|x' - \zeta| \leq f|x - \zeta|^2$$

Proof. Without loss of generality, by dividing everything by ζ, we can assume $\zeta = 1$. We then have the following equality:

$$\frac{x' - 1}{(x-1)^2} = \frac{(f x^{f-1}(x-1) - x^f + 1)(x-1)^{-2}}{f x^{f-1}} = \frac{\sum_{k=1}^{f-1} k x^{k-1}}{f x^{f-1}}$$

Applying the triangular inequality gives:

$$\left| \frac{x' - 1}{(x-1)^2} \right| \leq \frac{f(f-1)\max(1, |x|^{f-1})}{2f|x|^{f-1}} \leq \frac{1}{2} f \max(1, |x|^{1-f}).$$

We can conclude by noticing that $(1 - \frac{1}{2f})^{-f} \leq (1 - 1/6)^{-3} < 2$. ∎

For $f \geq 128$, it is now easy to show that the sequence converges towards $\exp(2i\pi/f)$; the finite number of remaining cases are easily done by direct computations.

3.1.2 A bound on the loss when iterating unitary matrices

We now show the following elementary lemma on the iterations of matrix-vector computations, which states that the error made when computing chained matrix-vector multiplications can be controlled.

Lemma 3.1.2. *Let A_i be a family of k unitary matrices. Suppose that for each of these matrices A_i there exists an algorithm $\mathcal{A}_i$ that given some vector x, outputs $A_i x$ within a certain vector of errors e such that $\|e\| \leq \epsilon \|x\|$ with $\epsilon \leq \frac{1}{2k}$. Then, the algorithm which computes $(\prod_i A_i)x$ by composing the algorithms $\mathcal{A}_i$ returns $(\prod_i A_i)x$ within an error vector e such that $\|e\| \leq 2k\epsilon\|x\|$.*

Proof. Let $B = \prod_{i=2}^{k} A_i$ and $Bx + e'$ the error committed using the algorithms $\mathcal{A}_i$. The algorithm $\mathcal{A}_1$ outputs $A_1(Bx + e') + e$, so that the error committed towards $A_1 Bx$ is

$$\|A_1(Bx + e') + e - A_1 Bx\| \leq \|e'\| + \|e\| \leq \|e'\| + \epsilon\|Bx + e\|$$

We now prove by induction that this error is less than $\left((1+\epsilon)^k - 1\right)\|x\|$ with:

$$\begin{aligned}\|e'\| + \epsilon\|Bx + e\| &\leq \left((1+\epsilon)^{k-1} - 1\right)\|x\| + \epsilon\left(\|x\| + \left((1+\epsilon)^{k-1} - 1\right)\|x\|\right) \\ &= \left((1+\epsilon)^k - 1\right)\|x\|.\end{aligned}$$

The case $k = 1$ is immediate and $(1+\epsilon)^k - 1 < 2k\epsilon$ for $\epsilon < \frac{1}{2k}$ finishes the proof.

■

3.1.3 Analysis of the Discrete Fourier transform

We now show how to efficiently compute a close approximation of a Fourier transform. Indeed, the fast Fourier transform on 2^n points correspond to a product of n unitary matrices, so that we can get p bits of precision using a precision in $O(p + \log n)$ by Lemma 3.1.2. Using this, we obtain an algorithm to multiply integers with B bits with complexity $O(B/w \cdot \log(B/w)) = O(B)$.

Bluestein's algorithm [19] for Chirp-Z transform reduces discrete Fourier transform in any size to the computation of fast Fourier transform over power-of-two so that the same holds. Recall that Inverse Fourier transform can also be computed from a discrete Fourier transform.

All in all, we can evaluate the corresponding Fourier isomorphism and its inverse:

$$\mathbf{R}[x]/(\Phi_f) \cong \mathbf{C}^{\varphi(f)/2}$$

with limited loss in precision.

The complexity of this computation is a $O(np + n \log n \cdot p/w) = O(np)$ for $p = \Omega(w + \log n)$ with $n = \varphi(f)$. Indeed it breaks down as:

- Write the coefficients as polynomials with register-size coefficients and compute their Fourier transform with a cost of $O(np)$
- Compute $O(p/w)$ convolutions with Fourier transforms of size $O(n)$
- Compute the inverse transform and propagate the carries for a running time of $O(np)$.

(A modular implementation is probably faster if n is not tiny.)

In the general case, one would have to precompute the roots and use product and remainder trees [123].

3.2 HOUSEHOLDER ORTHOGONALIZATION

The Householder orthogonalization algorithm transforms a complex matrix A into a product of QR, with Q unitary and R upper-triangular. Q is formed as a product of unitary reflections, which are all of the type $\mathrm{Id} - 2v\overline{v}^t$ for certain vectors $\|v\| = 1$.

The vector v corresponding to the first symmetry is chosen so that the first column of R has only its first coordinate to be non-zero. The algorithm then applies this unitary operation to the matrix A and recursively orthogonalize the bottom-right of this new matrix.

More precisely, denote by a the first column of the matrix A. As such, the first column of R will be the vector

$$r = \left(-\|a\| \cdot \frac{a_1}{|a_1|}, 0, \dots, 0 \right)^t,$$

with the quotient $\frac{a_1}{|a_1|}$ set to 1 if $a_1 = 0$. Then with $v = \frac{a-r}{\|a-r\|}$ and $Q = \mathrm{Id} - 2v\overline{v}^t$, we have that:

$$Qa = a - 2\frac{(a-r)\overline{(a-r)}^t a}{\|a-r\|^2} = a - \frac{2(\|a\|^2 - \overline{r}^t a)}{\|a-r\|^2}(a-r)$$

We now use the fact that $\overline{a}^t r \in \mathbf{R}$ and $\|r\| = \|a\|$ to get:

$$2(\|a\|^2 - \overline{r}^t a) = \|a\|^2 - \overline{r}^t a - \overline{a}^t r + \|r\|^2 = \|a - r\|^2$$

so that $Qa = r$.

The sign in the definition of r implies that $\|a - r\| \geq \|a\|$ so that we can compute v with the precision used to handle a.

If we use $p > \omega(\log d)$ bits of precision, we can multiply by $\mathrm{Id} - 2v\overline{v}^t$ with a relative error of $O(d2^{-p})$. Using Lemma 3.1.2, since we are performing d symmetries, each column is computed with a relative error of at most a $O(d^2 2^{-p})$. Hence, with $\hat{Q}$ the matrix output by the algorithm, each column

of $\overline{Q}^t A$ has a relative error of $O(d^2 2^{-p})$ with respect to the computed R. This implies that there exists a matrix A' where each column is A within a relative error of $O(d^2 2^{-p})$, and whose R-factor in the QR decomposition is the returned R. Remark that the returned $R_{i,i}$ may not be real. While this is usually not a problem, R has to be multiplied on the left by a diagonal unitary matrix to obtain *the* QR-decomposition.

We define the conditional number of A as $\kappa(A) = \|A\|\|A^{-1}\|$. We can bound the stability of the QR decomposition [157]:

Theorem 3.2.1. *Given a matrix A, let R be the R-factor of its QR decomposition. For the matrix $A + \delta A$, let $R + E$ be the R-factor of its QR decomposition. Then:*

$$\|E\| \leq 3\kappa(A)\|\delta A\|$$

provided that $\kappa(A)\frac{\|\delta A\|}{\|A\|} < 1/10$.

Proof. Let $A = QR$ be the QR-decomposition. Without loss of generality, we assume $\|A\| = 1$. For a technical reason, we study the problem with δA a linear function where $\delta A(1)$ is the wanted matrix, which means that other quantities such as E are also functions.

We now obtain:

$$\overline{(A+\delta A)}^t (A + \delta A) = \overline{A}^t A + \overline{\delta A}^t A + \overline{A}^t \delta A + \overline{\delta A}^t \delta A$$

which is equal to:

$$\overline{(R+E)}^t (R + E) = \overline{R}^t R + \overline{E}^t R + \overline{R}^t E + \overline{E}^t E$$

so we deduce:

$$\overline{E}^t R + \overline{R}^t E + \overline{E}^t E = \overline{\delta A}^t A + \overline{A}^t \delta A + \overline{\delta A}^t \delta A.$$

We multiply by $\overline{A}^{-t}$ on the left and A^{-1} on the right:

$$\overline{A}^{-t}\overline{E}^t\overline{Q}^t + QEA^{-1} + \overline{A}^{-t}\overline{E}^t EA^{-1} = \overline{A}^{-t}\overline{\delta A}^t + \delta A A^{-1} + \overline{A}^{-t}\overline{\delta A}^t \delta A A^{-1}.$$

With $\rho = \|EA^{-1}\|$ and $\epsilon = \|\delta A A^{-1}\|$, we take the norm and get the inequality:

$$\rho - \rho^2 \leq 2\epsilon + \epsilon^2$$

so that for $\epsilon < 1/10$ we have $\rho \leq 3\epsilon$ if $\rho < 1/2$.

We now have to exclude the case $\rho > 1/2$, which we do with a topological argument. It is clear from the algorithm that the QR-decomposition is continuous over invertible matrices. Since

$$\|A^{-1}(A + \delta A(t)) - \mathrm{Id}\| \leq \|A^{-1}\|\|\delta A(t)\| < 1/2$$

for $0 \leq t \leq 1$, we have that $A + \delta A$ is invertible and therefore ρ is continuous over $[0;1]$. As $\rho(0) = 0$ and $\rho([0;1])$ is connex, we get $\rho(1) < 1/2$.

Finally $\|E\| \leq \|EA^{-1}\|\|A\| = \rho$ gives the result. ∎

Combining these results, we get:

Theorem 3.2.2. *Given a matrix A, we can compute the R-factor of its QR decomposition in time*

$$O\left(\frac{d^3 p}{w} + d^3 + d^2 p\right)$$

with a relative error of

$$O\big(\kappa(A) d^2 2^{-p}\big)$$

if this is smaller than a constant.

We can, of course, decrease the 3 in the exponent to a few matrix multiplications using aggregated Householder transformations and a divide-and-conquer algorithm, see [79, Subsection 18.4]. This is also at the end of the appendix.

3.3 SIZE-REDUCTION

We first consider the size-reduction for unitriangular matrices (i.e. upper triangular matrices with ones on the diagonal). Such a matrix A is said to be size-reduced if both A and A^{-1} are small.

Lemma 3.3.1. *Let A be a unitriangular matrix of dimension d with coefficients in $\mathbf{K} = \mathbf{Q}[\zeta_f]$, such that its coefficients in the power basis are bounded in absolute value by 1. Then $\|A\| \leq d n^{3/2}$ and $\|A^{-1}\| = (2n)^{O(d)}$ with $n = \varphi(f)$.*

Proof. It is clear that $\|A_{i,j}\| \leq \sqrt{nf} \leq n^{3/2}$ so that $\|A\| \leq d n^{3/2}$. Now let x be a column of A^{-1}. Consider a i which maximizes $\|x_i\|(2n^{3/2})^i$. Then we have

$$1 \geq \|(Ax)_i\| \geq \|x_i\| - \sum_{j>i} \|A_{i,j}\| \|x_j\| \geq \|x_i\| \left(1 - \sum_{j>i} \frac{n^{3/2}}{(2n^{3/2})^{j-i}}\right) > \|x_i\|/3$$

and we obtain $\|x_i\| \leq 3$ which gives $\|x\| \leq 3(2n^{3/2})^{d-1}\sqrt{d}$. ∎

We can finally prove our size-reduction theorem:

Theorem 3.3.1. *Let A be a matrix of dimension d with coefficients in $\mathbf{K} = \mathbf{Q}[\zeta_f]$, and $n = \varphi(f)$. We are given p, where $\|A\|, \|A^{-1}\| \leq 2^p$ and also $\sqrt{n \log n \log\log n} + d \log n < p$. In time $O\big(d^3 n p / w + d^2 p n \log d\big)$, we can find an integral triangular matrix U with $U_{i,i} \in \mathscr{O}_{\mathbf{K}}^{\times}$ and a matrix $R + E$ such that $\|E\| \leq 2^{-p}$, with R the R-factor of the QR decomposition of AU and*

$$\kappa(AU) \leq \left(\frac{\max_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)^{1/n} 2^{O\left(\sqrt{n \log n \log\log n} + d \log n\right)}.$$

We also have $\|U\| \in 2^{O(p)}$

Proof. In the canonical basis of $\mathbf{K}$ repeated d times, A corresponds to a $d \times d$ block matrix, where each block is a diagonal complex matrix of size $n/2 \times n/2$, so that the QR decomposition can be obtained from $n/2$ complex QR decompositions of dimension d. We can transform into (and from) this basis at a cost of $O(d^2 pn)$; and the same technique can be used with the size-reduction algorithm.

The algorithm computes R', the R-factor of the QR decomposition of A. Then we use Algorithm 20 on R' which returns a U, and the algorithm returns U and $R'U$.

We have that

$$\|AU\| \leq d \sum_i \|R_{i,i}\| \leq d^2 \|A\|$$

so that $\|U\| \leq \|A^{-1}\| \|AU\| \leq d^2 2^{2p}$. As a result, we can use a precision of $O(p)$ bits.

Let D be the diagonal of R. We have $\kappa(AU) = \kappa(R) \leq \kappa(D)\kappa(D^{-1}R)$. The reduction with units guarantees that

$$\kappa(D) \leq \left(\frac{\max_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})} \right)^{1/n} 2^{O\left(\sqrt{n \log n \log\log n}\right)}.$$

The previous lemma gives $\kappa(D^{-1}R) = 2^{O(d \log n)}$. ∎

3.3.1 On the reduction of well-conditioned matrices

We finish this section with properties of lattices represented by a well-conditioned matrix. The following easy theorem indicates that if we want to reduce the lattice generated by A, we can always truncate the matrix and work with precision only $O(\log(\kappa(A)))$. The transition matrix which will be computed by the algorithm also needs at most this precision. Up to an irrelevant (small) quantity, this is of course a $O\left(\log\left(\frac{\max_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)/n\right)$.

Theorem 3.3.2. *Let $A, \delta A$ and U an integer matrix such that $\|AU\| \leq \kappa \|A\|$, $\kappa(AU) \leq \kappa$ and*

$$\frac{\|\delta A\|}{\|A\|} \leq \frac{\epsilon}{3\kappa^3}$$

with $\epsilon < 1/4$ and $\kappa \geq \kappa(A)$. Let R be the R-factor of the QR-decomposition of AU and $R + E$ be the one of $(A + \delta A)U$. Then $\|U\| \leq \kappa^2$ and

$$\frac{\|E\|}{\|A\|} \leq \epsilon.$$

Proof. First $\|U\| \leq \|A^{-1}\| \|AU\| \leq \kappa \|A^{-1}\| \|A\| \leq \kappa^2$. Then $\|U\| \geq 1$ since it is integral so that $1 \leq \|A^{-1}AU\| \leq \|A^{-1}\| \|AU\|$ and $\|AU\| \geq \frac{1}{\|A^{-1}\|} = \frac{\kappa(A)}{\|A\|}$. We deduce:

$$\frac{\|\delta AU\|}{\|AU\|} \leq \frac{\epsilon}{3\kappa^2}$$

and applying the stability theorem we get:

$$\frac{\|E\|}{\|AU\|} \leq \frac{\epsilon}{\kappa}.$$

Using the lower bound on $\|AU\|$ finishes the proof. ■

In all LLL algorithms, $\max_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})$ is non-increasing with respect to the round number and $\min_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})$ is non-decreasing so that we can use the theorem for all U where AU is size-reduced with

$$\kappa \leq \left(\frac{\max_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i N_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)^{1/n} 2^{\mathrm{O}\left(\sqrt{n \log n \log\log n} + d \log n\right)}.$$

Heuristically, for random lattices, we have $\|U\| \lesssim \sqrt{\kappa(A)}$ and $\kappa(AU)$ depends only on the dimension so a truncation of the R-factor of the QR-decomposition of A with error roughly $\|A\|/\kappa(A)$ is enough. The precision needed is therefore on the order of $2\log(\kappa(A))$.

4

UNIT ROUNDING FOR ARBITRARY CYCLOTOMIC FIELDS

We recall that we want to prove the following general rounding theorem:

Theorem 4.0.1. *Let* $\mathbf{L}$ *be the cyclotomic field of conductor* f*. There is a quasi-linear randomized algorithm that given any element in* $x \in (\mathbf{R} \otimes \mathbf{L})^\times$ *finds a unit* $u \in \mathscr{O}_{\mathbf{L}}^\times$ *such that for any field embedding* $\sigma : \mathbf{L} \to \mathbf{C}$ *we have*

$$\sigma\left(xu^{-1}\right) = 2^{O\left(\sqrt{f \log f}\right)} N_{\mathbf{L}/\mathbf{Q}}(x)^{\frac{1}{\varphi(f)}}.$$

4.1 SETTING.

Let us consider an integer f and take its prime decomposition $f = \prod_{i=1}^{r} p_i^{e_i}$. We set $q_i = p_i^{e_i}$ and we fix the cyclotomic field $\mathbf{L} = \mathbf{Q}[\zeta_f]$ of conductor f. Classically, the Galois group of $\mathbf{L}$ is equal to $G = {(\mathbf{Z}/f\mathbf{Z})^\times}/{\{-1, 1\}}$, whose elements are the σ_α, sending ζ_f to ζ_f^α for any $\alpha \in G$.

4.2 CYCLOTOMIC UNITS AND THEIR GENERATORS.

The cyclotomic units are defined as all the products of $\pm\zeta_f$ and $\zeta_f^a - 1$ which are units. We let $\mathcal{Q}$ be the set of the 2^r possible products of the q_i.

A standard theorem of [101, Lemma 2.2] reduces the number of generators of the cyclotomic units:

Theorem 4.2.1. *The cyclotomic units are all the products of* $\pm\zeta_f$ *and* $G \cdot (\zeta_f^a - 1)$ *which are units, when* a *runs through* $\mathcal{Q}$*.*

Proof. Let $a \in \mathbf{Z}$, and define k to be the product of all the q_i dividing a, so that by construction $k \in \mathcal{Q}$. Now, we have:

$$1 - \zeta_f^a = \prod_{i=0}^{\frac{a}{k}-1} 1 - \zeta_f^{k+\frac{ifk}{a}}.$$

Let $p_j | k + \frac{ifk}{a}$. Remark that $p_j | \frac{fk}{a}$, so that $p_j | k$, and by definition of k we have $q_j | k$. We have therefore $q_j | \frac{fk}{a}$ and hence $\zeta_f^{k+ifk/a} - 1 \in \pm G \cdot \zeta_f^k - 1$. ∎

Theorem 4.2.2. *Let χ be an even Dirichlet character of conductor $c \mid f$ with $c > 1$ and $e \in X$. Then if c and e are coprime, then*

$$|\chi(Log(\zeta_f^e - 1))| = \frac{\varphi(e)\sqrt{c}}{2\ln(2)} \left(\prod_{\substack{i \\ p_i \mid \frac{f}{e}}} |1 - \chi(p_i)| \right) |L(1,\chi)|$$

else it is 0.

Proof. If $\gcd(c,e) > 1$, we have $\sum_{\alpha \in (\mathbf{Z}/\gcd(c,e)\mathbf{Z})^\times} \chi(\alpha) = 0$ so the result is zero. We therefore assume for now on that c and e are coprime.

We first compute:

$$\prod_{\substack{\beta \in G \\ \beta = 1 \bmod c}} 1 - \zeta_{\frac{f}{e}}^{\beta}.$$

Let $p_i \mid \frac{f}{ec}$ and $p_i \mid c$. Then:

$$\begin{aligned} \prod_{\substack{\beta \in G \\ \beta = 1 \bmod c}} 1 - \zeta_{\frac{f}{e}}^{\beta} &= \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cp_i}} \prod_{j=0}^{p_i - 1} 1 - \zeta_{\frac{f}{e}}^{\beta} \zeta_{p_i}^{j} \\ &= \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cp_i}} 1 - \zeta_{\frac{f}{e}}^{p_i \beta}. \end{aligned}$$

In the same way, we have if $p_i \mid \frac{f}{e}$ and $p_i \nmid c$, with $r^{-1} = \frac{f}{eq_i} \bmod p_i$:

$$\begin{aligned} \prod_{\substack{\beta \in G \\ \beta = 1 \bmod c}} 1 - \zeta_{\frac{f}{e}}^{\beta} &= \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cq_i}} \prod_{\substack{j=0 \\ j \neq -r \bmod p_i}}^{q_i - 1} 1 - \zeta_{\frac{f}{e}}^{\beta} \zeta_{q_i}^{\beta j} \\ &= \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cq_i}} \frac{1 - \zeta_{\frac{f}{e}}^{\beta q_i}}{1 - \zeta_{\frac{f}{e}}^{\beta (q_i - \frac{rf}{e})/p_i}} \\ &= \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cq_i}} \frac{1 - \zeta_{\frac{f}{eq_i}}^{\beta}}{1 - \zeta_{\frac{f}{eq_i}}^{\frac{\beta}{p_i}}}. \end{aligned}$$

In case $p_i \mid e$, we have $q_i \mid e$ and therefore

$$\prod_{\substack{\beta \in G \\ \beta = 1 \bmod c}} 1 - \zeta_{\frac{f}{e}}^{\beta} = \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cq_i}} \left(1 - \zeta_{\frac{f}{e}}^{\beta}\right)^{\varphi(q_i)}.$$

We can now compute our sum:

$$\begin{aligned} \sum_{\alpha \in G} \chi(\alpha) \log(|\zeta_f^{e\alpha} - 1|) &= \sum_{\alpha \in (\mathbf{Z}/c\mathbf{Z})^\times / \{-1,1\}} \chi(\alpha) \log\left(\left| \sigma_\alpha \left(\prod_{\beta \in G \beta = 1 \bmod c} \zeta_{\frac{f}{e}}^{\beta} - 1 \right) \right| \right) \\ &= \varphi(e) \left(\prod_{\substack{i \\ p_i \mid \frac{f}{e} \\ p_i \nmid c}} 1 - \chi(p_i) \right) \sum_{\alpha \in (\mathbf{Z}/c\mathbf{Z})^\times / \{-1,1\}} \chi(\alpha) \log(|\zeta_c^{\alpha} - 1|). \end{aligned}$$

We finish by the standard computation ([163, Theorem 4.9]) of the term on the right with the Gauss sum: $\tau = \sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times} \overline{\chi}(\alpha)\zeta_c^\alpha$:

$$\begin{aligned}\sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times} \overline{\chi}(\alpha)\ln(|\zeta_c^\alpha - 1|) &= \sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times} \overline{\chi}(\alpha)\ln(1-\zeta_c^\alpha)\\ &= \sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times}\sum_{k=1}^{\infty}\overline{\chi}(\alpha)\frac{\zeta_c^{\alpha k}}{k}\\ &= \sum_{i=1}^{\infty}\frac{\tau\chi(k)}{k} = \tau L(1,\chi)\end{aligned}$$

and $\tau\overline{\tau} = c$. ∎

Definition 4.2.1. *The* augmentation ideal *is the kernel of the form:* $(\sum_\alpha x_\alpha\sigma_\alpha \to \sum_\alpha x_\alpha)$ *over* $\mathbf{Z}[G]$.

With this definition we can complete the description of the cyclotomic units:

Theorem 4.2.3. *[101, Lemma 2.4] The cyclotomic units are generated by:*

- *The pair* $\pm\zeta_f$,
- *the* $G\cdot\zeta_f^a - 1$ *for all* $a \in \mathcal{Q}$ *such that* $\frac{f}{a}$ *is not prime power,*
- *the orbit of* $\zeta_f^{f/q_i} - 1$ *by the action of the augmentation ideal.*

Proof. Note first that for any $a \in \mathcal{Q}$, $(1-\sigma_\alpha)\cdot(\zeta_f^a - 1) \in \mathscr{O}_{\mathbf{L}}$. Next, we prove that an element u generated by the $\zeta_f^a - 1$ is a unit if $N_{\mathbf{L}/\mathbf{Q}}(u) = 1$. We remark that

$$\varphi(f)\cdot u = N_{\mathbf{L}/\mathbf{Q}}(u)\left(\left(\sum_\alpha 1-\sigma_\alpha\right)\cdot u\right) = \left(\sum_\alpha 1-\sigma_\alpha\right)\cdot u$$

so that it is a unit. The converse is clear. Eventually a direct computation ensures that $N_{\mathbf{L}/\mathbf{Q}}(1-\zeta_f^a)$ is $p_i^{\varphi(a)}$ if $a = f/q_i$ and 1 else using the equations at the beginning of the proof of Theorem 4.2.2. ∎

4.3 CONSTRUCTION OF AN "ORTHOGONAL" BASIS

We now define the family $(b_i)_{1\le i\le|\mathcal{Q}|}$ by setting $b_i = \mathrm{Log}(\zeta_f^a - 1)$ where the $a \in \mathcal{Q}$ are taken in decreasing order. We can define some Gram-Schmidt orthogonalization on this family with the relations:

$$b_i^* = b_i - \sum_{j<i}\frac{\left\langle b_i, b_j^*\right\rangle}{\left\langle b_j^*, b_j^*\right\rangle}b_j^* = b_i - \sum_{j<i} b_i b_j^*(b_j^*)^\dagger$$

where the dagger is the Moore-Penrose pseudo-inverse. As such, $\chi(b_i) = \chi(b_i^*)$ if $\chi(b_j^*) = 0$ for all $j < i$, and is equal to zero elsewhere. As $L(1,\chi) \neq$

0, we have for all $\chi \neq 1$ that $\chi(b_i^*) \neq 0$ iff $\mathrm{rad}(\frac{f}{e})|c|\frac{f}{e}$ where c is the conductor of the character χ. Furthermore, in this case, the term $\prod_{p_i|\frac{f}{e}}(1-\chi(p_i))$ is one. We can now give our decoding algorithm, assuming again that the cyclotomic units have a finite index:

Proof of the rounding theorem for arbitrary cyclotomic fields. We let $b_i = \mathrm{Log}(\zeta_f^e - 1)$ and recall that for all χ with conductor not coprime with e we have $\chi(b_i) = 0$. We remark that if $\frac{f}{e}$ is a prime power, we have $b_i^* = b_i$ and as a result $\|b_i^*\|_\infty \leq \log(\frac{f}{e})$. Also, we have for all i that $\|b_i^*\| \leq \|b_i\| = \mathrm{O}\left(\sqrt{\varphi\left(\frac{f}{e}\right)}\right)$ using the same technique. The algorithm consists in using Babai reduction with our generating family, with the modification described above to round with respect to the augmentation ideal when we have to. More precisely, for any $y \in \mathbf{Z}[G]b_i^*$, we compute z a randomized rounding of y/b_i^* in the same way as in the previous section. If $\frac{f}{e}$ is a prime power, the rounding is $z - \sum_\alpha z_\alpha \sigma_1$, else it is z. If $|\sum_\alpha z_\alpha| \geq \frac{\sqrt{\frac{f}{e}}}{\log(\frac{f}{e})}$ in case where $\frac{f}{e}$ is a prime power, we restart the rounding. We then continue in the same way with $i-1$. The analysis is as before. The randomized rounding produces an error with subgaussian coordinates with parameter $\mathrm{O}\left(\sqrt{\sum_{e\in X}\varphi(\frac{f}{e})}\right) = \mathrm{O}(\sqrt{f})$. The correction for the prime power adds an error bounded by $\sum_i \log(q_i)\sqrt{q_i}/\log(q_i) = O(\sqrt{f})$. Hence, the bound on the output holds. The running time is quasi-linear since we can work at each step with the ring

$$\mathbf{Z}\left[\left(\mathbf{Z}\Big/\left(\frac{f}{e}\right)\mathbf{Z}\right)^\times\right].$$

■

5

GENERALIZATION OF THE SYMPLECTIC DESCENT.

In Chapter 5, we described how to descend the symplectic structure when $\mathbf{L} = \mathbf{K}[X]/(X^d + a)$, that is for Kummer-like extensions. We show here the general case, that is when $\mathbf{L} = \mathbf{K}[X]/f(X)$. We first give a simple construction which recovers the one given above but has losses in the general case; and then describe a general construction without losses.

5.1 THE DUAL INTEGER CONSTRUCTION

We have the following lemma, proved in [127, Chapter III, Proposition 2.4]:

Lemma 5.1.1. *Let $a_i = X^i$ and $\sum_i b_i Y^i = \frac{f(Y)}{Y-X}$. Then $\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(a_i b_j / f'(X))$ is equal to 1 if $i = j$ and 0 else.*

This suggests taking as a $\mathbf{K}-$basis for $\mathbf{L}^2$ the $(a_i, 0)$ followed by the $(0, b_i)$. With the notations of Section 5.4, we now define $J'_{\mathbf{L}}$ as

$$\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(J_{\mathbf{L}}/f'(X)).$$

It follows from the lemma that in our basis, this is represented by the Darboux matrix:

$$\begin{pmatrix} 0 & \mathrm{Id}_d \\ -\mathrm{Id}_d & 0 \end{pmatrix}$$

and, as usual, we can reverse the order of the second part of the basis to obtain the wanted matrix.

We can convert efficiently a number $z \in \mathbf{L}$ in the basis of b_i. Clearly, the coefficients are given by all the $\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(z/f'(X) \cdot X^i)$. We then simply evaluate $z/f'(X)$ on all roots of f using a remainder tree, and follow by a Vandermonde matrix-vector multiplication, which is also a multipoint evaluation [123]. In particular, we do not need to compute the b_i.

There is however a loss with this basis: the algorithm tries to minimize the size of the coefficients in our basis of $\mathbf{L}^2$ instead of the canonical norm.

5.2 THE ORTHOGONAL CONSTRUCTION

We want to build an orthogonal $\mathbf{R} \otimes \mathbf{K}$-basis of $\mathbf{R} \otimes \mathbf{L}$. We assume for simplicity (only) that $\mathbf{L}$ (and therefore $\mathbf{K}$) is a totally real field. Hence, with $\mathbf{K} = \mathbf{Q}[Y]/g(Y)$, we have that all roots r_i of g are real, and when we evaluate all coefficients of f on r_i, the resulting polynomial has real roots $r_{i,j}$.

We then define the j-th element of the basis as being the element of $\mathbf{L}$ which, when we evaluate on $(X - r_{i,k}, Y - r_i)$, we obtain 1 if $j = k$ and 0 else. This is clearly an orthogonal basis for the canonical norm, and in this case, it is also its dual. Hence, using twice this basis leads again to the Darboux matrix for $J'_{\mathbf{L}} = \mathrm{tr}_{\mathbf{L}/\mathbf{K}}(J_{\mathbf{L}})$. Exactly the same construction works for totally imaginary $\mathbf{K}$ (and therefore $\mathbf{L}$).

The general case can be done in the same way, by taking care of ramified places.

COLOPHON

Final Version as of November 18, 2019Ṫhomas Espitau.