RÉSEAUX EUCLIDIENS ET CRYPTOGRAPHIE - MASTER 2, UNIVERSITÉ DE RENNES

ALEXANDRE WALLET

NOTATIONS ET RAPPEL D'ALGÈBRE LINÉAIRE

Tout au long de ce cours, et sauf mention contraire, l'espace vectoriel ambiant sera $\mathbb{R}^m$. Le vecteur des coordonnées de x dans une base d'un sous-espace contenant x sera noté en minuscule grasse, par exemple $\mathbf{x} = (x_1, \dots, x_m)$. Si en cryptographie, il est coutume de considérer des vecteurs en ligne [1], mais on les préférera ici en colonne. En particulier, étant donné $n \leqslant m$ vecteurs $v_1, \dots, v_n$, on notera en majuscules grasses, par exemple $\mathbf{V}$, la matrice à m lignes et n colonnes dont les colonnes sont les coordonnées des v_i dans une base, ou encore $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{m \times n}$.

On rappelle que toute forme quadratique définie positive q correspond de manière unique à une forme bilinéaire symétrique définie positive $\langle , \rangle$ par

$$\langle x, y \rangle = \frac{1}{2}(q(x+y) - q(x) - q(y)) \quad \text{et} \quad q(x) = \langle x, x \rangle.$$

On notera indifféremment $q(x) = q(\mathbf{x})$ et $\langle x, y \rangle = \langle \mathbf{x}, \mathbf{y} \rangle$. Un espace euclidien est un espace vectoriel réel de dimension finie muni d'une forme quadratique définie positive, et il est donc équipé naturellement d'une norme $\|x\|_q^2 := q(x) = \langle x, x \rangle$. Sauf mention contraire, le seul espace euclidien qu'on utilisera ici est $\mathbb{R}^m$ muni du produit scalaire canonique $\langle , \rangle$, ou de manière équivalente, de la norme euclidienne usuelle. Leurs définitions sont :

$$\langle x, y \rangle = \sum_{i \leqslant m} x_i y_i = \mathbf{x}^t \mathbf{y} \quad \text{et} \quad \|x\|^2 := \sum_{i \leqslant m} x_i^2 = \mathbf{x}^t \mathbf{x},$$

où les x_i, y_i sont les coordonnées de x, y dans la base canonique de $\mathbb{R}^m$.

Soit $n \leqslant m$ et une famille libre $(b_i)_{1 \leqslant i \leqslant n}$ de $\mathbb{R}^m$, engendrant un sous-espace $V \subset \mathbb{R}^m$. Tout $x \in V$ s'écrit de manière unique $x = \sum_{i \leqslant n} x_i b_i$, pour $x_1, \dots, x_n$ réels. Si $\mathbf{B}$ est la matrice des $(b_i)_i$ dans la base canonique de $\mathbb{R}^m$, et $\mathbf{x} = (x_1, \dots, x_n)$ les coordonnées de x dans la base $(b_i)_i$, on a $x = \mathbf{B}\mathbf{x}$.

1. RÉSEAUX EUCLIDIENS

On rappelle qu'on muni $\mathbb{R}^m$ de la norme euclidienne standard. Pour ce cours orienté vers les applications en cryptographie, on utilisera la définition suivante d'un réseau euclidien.

Définition 1.1. Un réseau euclidien est un sous-groupe *discret* de $\mathbb{R}^m$.

Soit $\mathscr{L} \subsetneq \mathbb{R}^m$ un réseau euclidien. On peut montrer qu'il existe une famille $\mathbb{Z}$-libre maximale $(b_i)_{1 \leqslant i \leqslant n}$ dans $\mathscr{L}$ telle que $\mathscr{L} = \mathbb{Z}b_1 \oplus \cdots \oplus \mathbb{Z}b_n$. Cette famille étant libre, c'est une base du sous-espace $V = \mathbb{R}b_1 \oplus \ldots \oplus \mathbb{R}b_n$ de dimension n. Comme elle est maximale dans $\mathscr{L}$, c'est-à-dire qu'ajouter un vecteur de $\mathscr{L}$ à cette famille la rend $\mathbb{Z}$-linéairement dépendante, on l'appelle aussi *base* du réseau $\mathscr{L}$.

1. Ils apparaissent d'ailleurs en ligne dans le texte aussi, bof. Autrement dit, $\mathbf{x}^t$ est un vecteur ligne.

Définition 1.2. L'entier n est commun à toutes les bases de $\mathscr{L}$, et on l'appelle le *rang* de $\mathscr{L}$ et on note $n = \mathrm{rk}\,\mathscr{L}$. Lorsque $m = n$ on dit parfois que le réseau est de rang plein.

Remarque 1.3. Toutes les bases de $\mathbb{R}^n$ engendrent des réseaux en restreignant les scalaires à $\mathbb{Z}$. Toutes les bases d'un réseau de rang n fixé engendrent un espace vectoriel de dimension n en étendant les scalaires à $\mathbb{R}$.

Etant donnée une famille libre $(b_i)_{i\leqslant n}$ de $\mathbb{R}^m$, on note $\mathscr{L}(b_1,\dots,b_n) = \mathbb{Z}b_1 \oplus \cdots \oplus \mathbb{Z}b_n$ le réseau engendré par cette famille. Si on fixe une base $(e_i)_{1\leqslant i\leqslant m}$ de $\mathbb{R}^m$, et notons $\mathbf{B} = [\mathbf{b}_1,\dots,\mathbf{b}_n]$ la matrice colonne des (coordonnées des) $\mathbf{b}_i$ dans cette base, on notera aussi $\mathscr{L}(\mathbf{B})$ ou $\mathbf{B}\mathbb{Z}^n$. Les sous-réseaux de $\mathscr{L}$ sont les sous-groupes discrets de $\mathscr{L}$.

Exemple 1.4. On a une inclusion $2\mathbb{Z} \subset \mathbb{Z} \subset \frac{1}{2}\mathbb{Z}$ de réseaux euclidiens de $\mathbb{R}$. Cet exemple anodin souligne qu'avoir le même rang et une inclusion n'est pas suffisant pour avoir égalité entre des réseaux, a contrario du cas des espaces vectoriels. Pour tout entier $n \geqslant 1$, $\mathbb{Z}^n$ est un réseau euclidien de $\mathbb{R}^n$. L'ensemble D_n des vecteurs de $\mathbb{Z}^n$ dont la somme des coordonnées est paire est un sous-réseau strict de rang n de $\mathbb{Z}^n$. Si $(b_i)_{i\leqslant n}$ est une famille libre de $\mathbb{R}^m$, on a vu que $\mathscr{L}(b_1,\dots,b_n)$ est un réseau de rang n. Notons $\mathbf{B}$ la matrice des (b_i), et supposons de plus que les b_i ont des coordonnées entières. Alors $\mathscr{L}_q(\mathbf{B}) := \mathbf{B}\mathbb{Z}^n + q\mathbb{Z}^m$ est un réseau de rang m de $\mathbb{R}^m$. Pour tout entier $q > 1$, l'ensemble $\mathscr{L}_q^\perp(\mathbf{B}) = \{\mathbf{x} \in \mathbb{Z}^m \ : \ \mathbf{B}^t\mathbf{x} = 0 \bmod q\}$ est un réseau de rang m. Les réseaux comme $\mathscr{L}_q(\mathbf{B})$ et $\mathscr{L}_q^\perp(\mathbf{B})$ sont appelés des réseaux q-aires, et sont importants en cryptographie.

Tout comme dans le dernier exemple, on ne décrit pas toujours un réseau par l'une de ses bases. Par contre, on ne sait pas manipuler algorithmiquement un réseau sans l'une d'elles.

Exercice 1.5. Quelques exercices pour travailler son intuition :

— Montrer que les sous-groupes de $\mathbb{R}$ sont denses ou de la forme $\alpha\mathbb{Z}$ pour $\alpha \in \mathbb{R}$.

— L'ensemble $\mathbb{Z} + \sqrt{2}\mathbb{Z}$ est-il un réseau de $\mathbb{R}$? L'ensemble $\mathbb{Z} \oplus \sqrt{2}\mathbb{Z}$ est-il un réseau de $\mathbb{R}^2$?

— Montrer que tout réseau euclidien admet une base au sens défini ci-dessus.

— Soit $\mathscr{L}_1, \mathscr{L}_2$ deux réseaux de $\mathbb{R}^m$. Montrer que :

- $\mathscr{L}_1 + \mathscr{L}_2$ est un réseau de $\mathbb{R}^m$ tel que $\mathrm{rk}(\mathscr{L}_1 + \mathscr{L}_2) \geqslant \max(\mathrm{rk}\,\mathscr{L}_1, \mathrm{rk}\,\mathscr{L}_2)$;
- $\mathscr{L}_1 \cap \mathscr{L}_2$ est un réseau de $\mathbb{R}^m$ tel que $\mathrm{rk}(\mathscr{L}_1 \cap \mathscr{L}_2) \leqslant \min(\mathrm{rk}\,\mathscr{L}_1, \mathrm{rk}\,\mathscr{L}_2)$.

— Pourquoi $\mathscr{L}_q(\mathbf{B})$ est-il de rang m ?

1.1. **Bases des réseaux.**

Rappels sur le déterminant de matrices. Pour ce cours, on aura uniquement besoin de propriétés élémentaires. Le déterminant est défini pour les matrices *carrées*. On rappelle qu'une matrice $\mathbf{A} \in \mathbb{R}^{m\times m}$ est inversible si et seulement si ses colonnes forment une base de $\mathbb{R}^m$, si et seulement si son déterminant est non nul. Dans ce cas, la formule de Cramer pour l'inverse de $\mathbf{A}$ est

$$(\det \mathbf{A}) \cdot \mathbf{A}^{-1} = \mathrm{com}(\mathbf{A})^t,$$

où $\mathrm{com}(\mathbf{A})$ désigne la co-matrice de $\mathbf{A}$, c'est-à-dire la matrice dont l'entrée (i,j) est le déterminant de la matrice $\mathbf{A}$ où la ligne i et la colonne j ont été retirée. Pour toutes matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$, on a de plus

$$\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B}), \quad \det(\mathbf{A}^{-1}) = \frac{1}{\det \mathbf{A}}, \quad \text{et} \quad \det(\mathbf{A}^t) = \det(\mathbf{A}).$$

Quelques exemples instructifs. On peut développer l'intuition permettant de caractériser les bases d'un réseau facilement sur un exemple en dimension 2, avec quelques dessins [2]. Considérons l'exemple de $\mathbb{Z}^2$. L'une de ses bases est $e_1 = (1,0), e_2 = (0,1)$, correspondant à la matrice identité. D'autres bases sont données par exemple pas $e_1, u_a = (a,1)$: en effet, $u_a = ae_1 + e_2$ ou de manière équivalente $e_2 = u_a - ae_1$. Si $\mathbf{U}$ est la matrice de transformation de la base (e_1, e_2) à la base (e_1, u_a), on a

$$\mathbf{U} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \quad \text{et} \quad \mathbf{U}^{-1} = \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix}.$$

On remarque $\mathbf{U}$ et $\mathbf{U}^{-1}$ ont des entrées entières, et de plus que $\det \mathbf{U} = \det \mathbf{U}^{-1} = 1$. Soit maintenant $v_1 = (-1,3)$ et $v_2 = (-1,2)$, et $\mathscr{L} = \mathscr{L}(v_1, v_2)$. Bien que géométriquement, les vecteurs de départ semblent assez différents, on peut montrer aisément que $\mathscr{L} = \mathbb{Z}^2$: en effet, v_1, v_2 ont des coordonnées entières, et de plus, $e_2 = v_1 - v_2 \in \mathscr{L}$ et $e_1 = 2e_2 - v_2 \in \mathscr{L}$. La matrice de passage $\mathbf{U}$ entre (e_1, e_2) et (v_1, v_2) est

$$\mathbf{U} = \begin{bmatrix} -1 & -1 \\ 3 & 2 \end{bmatrix}, \quad \text{avec} \quad \mathbf{U}^{-1} = \begin{bmatrix} 2 & 1 \\ -3 & -1 \end{bmatrix}.$$

Encore une fois, l'inverse a ses coordonnées entières, et cette fois $\det \mathbf{U} = -1$. Regardons maintenant le réseau engendré par $w_1 = (1,-1)$ et $w_2 = (1,2)$. Il est clair que $\mathscr{L}(w_1, w_2) \subset \mathbb{Z}^2$, mais les matrices de passages sont

$$\mathbf{U} = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}, \quad \text{avec} \quad \mathbf{U}^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}.$$

En particulier, on ne peut pas écrire (e_1, e_2) comme combinaison linéaire *entière* de w_1, w_2 : $\mathscr{L}(w_1, w_2)$ est un sous-réseau strict de $\mathbb{Z}^2$. Ceci est encodé par le déterminant de la matrice de transformation, qui est ici $\det \mathbf{U} = 3$. Géométriquement, on se convainc rapidement que l'aire du parallélogramme décrit par (w_1, w_2) est plus grosse que celle générée par (e_1, e_2). On a en fait le résultat suivant.

Proposition 1.6. Soient $\mathscr{L} = \mathscr{L}(\mathbf{B})$ et $\mathscr{L}' = \mathscr{L}(\mathbf{B}')$ deux réseaux de rang n. Alors $\mathscr{L} = \mathscr{L}'$ si et seulement si il existe une matrice $\mathbf{U} \in \mathscr{M}_n(\mathbb{Z})$ telles que $\mathbf{B}' = \mathbf{B}\mathbf{U}$ et $|\det \mathbf{U}| = 1$.

Démonstration. Supposons que $\mathscr{L} = \mathscr{L}'$. Par définition, il existe deux matrices $\mathbf{U}, \mathbf{U}' \in \mathscr{M}_n(\mathbb{Z})$ telles que $\mathbf{B}' = \mathbf{B}\mathbf{U}$ et $\mathbf{B} = \mathbf{B}'\mathbf{U}'$. Ceci implique que $\mathbf{B}(\mathbf{I}_n - \mathbf{U}\mathbf{U}')$ est la matrice nulle. Comme $\mathbf{B}$ est de rang plein (puisque ses colonnes forment une famille libre), on a forcément $\mathbf{U}\mathbf{U}' = \mathbf{I}_n$. Puisque $\mathbf{U}$ et $\mathbf{U}'$ sont entières, leurs déterminants sont des entiers, dont le produit vaut 1 par ce qui précède.

Supposons maintenant que $\mathbf{B}' = \mathbf{B}\mathbf{U}$ pour une matrice entière $\mathbf{U}$ de déterminant 1. Ceci implique que $\mathscr{L}' \subset \mathscr{L}$, et aussi que $\mathbf{U}$ est inversible. On a alors $\mathbf{U}^{-1} = \mathrm{com}(\mathbf{U})^t$ d'après la formule de Cramer. Les entrées de $\mathrm{com}(\mathbf{U})$ sont des déterminants de sous-matrices de $\mathbf{U}$, dont les entrées sont entières, et on a de plus $\mathbf{B} = \mathbf{B}'\mathbf{U}^{-1}$. Les colonnes de $\mathbf{B}$ sont donc des combinaisons linéaires entières de celles de $\mathbf{B}'$ et donc $\mathscr{L} \subset \mathscr{L}'$. □

1.2. **Invariants fondamentaux.**

2. Seulement, je suis une tanche en tikz alors les dessins sont "en construction", faîtes les sur une feuille.

1.2.1. *Déterminant d'un réseau.*

Définition 1.7. Soit $(b_i)_{1\leqslant i\leqslant n}$ une famille libre de $\mathbb{R}^m$. La matrice de Gram de la famille $(b_i)_i$ est $\mathbf{G}_{(b_i)} = [\langle b_i, b_j\rangle]_{1\leqslant i,j\leqslant n}$.

Lorsque le contexte est clair, on omettra l'indice. Si $\mathbf{B}$ est la matrice colonne des b_i dans une base de $\mathbb{R}^m$, on a aussi $\mathbf{G} = \mathbf{B}^t\mathbf{B}$.

Définition 1.8. Soit une famille libre $(b_i)_{1\leqslant i\leqslant n}$ et $\mathscr{L}$ le réseau engendré. On appelle déterminant du réseau la quantité $\det\mathscr{L} = \sqrt{\det\mathbf{G}_{(b_i)}}$.

Le déterminant est bien indépendant de la base de $\mathscr{L}$. En effet, si $\mathbf{B},\mathbf{B}'$ sont deux bases (sous forme de matrices) pour $\mathscr{L}$, la proposition 1.6 assure l'existence de $\mathbf{U}\in\mathscr{M}_n(\mathbb{Z})$ de déterminant ±1 telle que $\mathbf{B}' = \mathbf{B}\mathbf{U}$. On a alors

$$\det(\mathbf{B}'^t\mathbf{B}') = \det(\mathbf{U}^t\mathbf{B}^t\mathbf{B}\mathbf{U}) = \det(\mathbf{U})^2\det(\mathbf{B}^t\mathbf{B}) = \det(\mathbf{B}^t\mathbf{B}).$$

Lorsque $\mathscr{L}$ est de rang m dans $\mathbb{R}^m$, et que $\mathbf{B}$ est la forme matricielle de l'une de ses bases, on a $\det\mathscr{L} = |\det\mathbf{B}|$.

Remarque 1.9.
En cryptographie, le déterminant d'un réseau est souvent appelé son *volume*. Cela vient du fait que chaque base $(b_i)_i$ d'un réseau définit un *domaine fondamental*

$$\mathscr{D}_{(b_i)_i} = \left\{\sum_i x_i b_i \ : x_i\in\left(-\frac{1}{2},\frac{1}{2}\right]\right\},$$

qui est un parallélépipèdede de volume fini $\det\mathbf{B}$. La proposition 1.6 dit que tous les domaines fondamentaux d'un réseau ont le même volume. On peut s'affranchir du choix d'une base et donner un sens rigoureux au volume d'un réseau euclidien de rang m en étudiant le groupe compact $\mathbb{R}^m/\mathscr{L}$, mais cela ne sera pas utile dans ce cours.

Remarque 1.10 (sur le volume d'une partie de $\mathbb{R}^m$). Il n'est pas dans le cadre de ce cours de refaire la théorie de la mesure pour définir convenablement la notion de volume. On se contentera de rappeler quelques notions intuitives, en fixant une forme simple : un "hypercube" de côté 1 dans $\mathbb{R}^m$. Informellement, toute partie bornée de $\mathbb{R}^m$ peut être approchée par des unions de cube (à la manière dont on construit l'intégrale de Riemann par exemple), donc les propriétés "raisonnables" du volume devraient provenir de ce qu'on observe sur des cubes. En particulier, le volume du cube ne devrait pas dépendre de "l'endroit" où il se trouve, c'et-à-dire être invariant par translation. Si on augmente tous les côtés du cube par le même facteur λ, le volume devrait augmenter en proportion λ^m, ce qu'on apelle l'homogénéité. Plus généralement, dilater chaque côté par un facteur λ_i devrait se traduire par une modification de $\prod_i\lambda_i$ du volume. Toujours informellement, ceci permet de comprendre le déterminant d'une transformation linéaire comme la façon dont elle modifie le volume d'un cube et de se convaincre que pour un cube C, on a $\mathrm{Vol}(TC) = \det(T)\,\mathrm{Vol}(C)$. Le déterminant d'un réseau se lit donc comme la façon dont il change la base canonique de $\mathbb{R}^m$.

Le critère utile suivant correspond à l'intuition que le domaine fondamental d'un sous-réseau devrait être plus volumineux que celui du réseau ambiant, lorsque la comparaison a du sens.

Proposition 1.11. Soient $\mathscr{L}'\subset\mathscr{L}$ deux réseaux de même rang. Alors on a $\det\mathscr{L}\leqslant\det\mathscr{L}'$, avec égalité si et seulement si $\mathscr{L}' = \mathscr{L}$.

Démonstration. Notons $\mathbf{B}, \mathbf{B}'$ des matrices pour des bases de $\mathscr{L}, \mathscr{L}'$. Par hypothèse, il existe une matrice entière $\mathbf{U}$ telle que $\mathbf{B}' = \mathbf{B}\mathbf{U}$. Par définition du déterminant, on a $\det \mathscr{L}' = |\det \mathbf{U}| \det \mathscr{L}$, avec $\det \mathbf{U} \in \mathbb{Z}$. Le cas d'égalité correspond à la Proposition 1.6. □

On peut montrer un résultat plus fort (preuve en TD).

Proposition 1.12. Soient $\mathscr{L}' \subset \mathscr{L}$ deux réseaux de même rang. Si $[\mathscr{L} : \mathscr{L}']$ est l'indice de $\mathscr{L}'$ dans $\mathscr{L}$ en tant que groupe abélien, on a

$$[\mathscr{L} : \mathscr{L}'] = \frac{\det \mathscr{L}'}{\det \mathscr{L}}.$$

Exemple 1.13. Toutes les hypothèses sont importantes dans les énoncés précédents :

— $\mathbb{Z}^2$ et $(2,0)\mathbb{Z} \oplus (0, \frac{1}{2})\mathbb{Z}$ ont le même déterminant et le même rang, mais sont distincts ;

— $\mathbb{Z}$ et $\mathbb{Z}^2$ ont le même déterminant mais pas le même rang.

— $\mathscr{L}' = (0,1)\mathbb{Z}$ est un sous-réseau de $\mathscr{L} = (4,0)\mathbb{Z} \oplus (0, \frac{1}{2})\mathbb{Z}$, avec $\det \mathscr{L}' \leqslant \det \mathscr{L}$.

En particulier, bien que le déterminant caractérise un réseau, il n'est pas suffisant pour les classifier.

1.2.2. *Minimum d'un réseau.* Un réseau euclidien étant discret, il existe nécessairement des vecteurs non nuls du réseau qui sont les plus courts possibles pour la norme ambiante. Notons $\mathscr{B}(x,r)$ la boule centrée en x et de rayon r pour cette norme. Lorsque la boule est centrée en 0, on note $\mathscr{B}(r)$. Pour tout ensemble fini S, on note $|S|$ son cardinal.

Définition 1.14. On appelle minimum d'un réseau $\mathscr{L}$ la quantité

$$\lambda_1(\mathscr{L}) = \min\{r > 0 \ : \ |\mathscr{B}(r) \cap \mathscr{L}| > 1\}.$$

Si le réseau ambiant est clair par contexte, on notera aussi simplement λ_1. Tout vecteur $v \in \mathscr{L} \setminus \{0\}$ tel que $\|v\| = \lambda_1(\mathscr{L})$ est appelé un plus court vecteur de $\mathscr{L}$.

1.3. **Le théorème de Minkowski.** Les deux invariants introduits, s'ils ne suffisent pas à caractériser un réseau complètement, sont cependant liés par le résultat fondamental suivant.

Théorème 1.15. Soit $\mathscr{L}$ un réseau de rang n. On a $\lambda_1(\mathscr{L}) \leqslant \sqrt{n} \cdot \det(\mathscr{L})^{1/n}$.

Le résultat est, du point de vue théorique, assez satisfaisant. L'exposant normalise le volume du réseau à une quantité "unidimensionnelle", et le théorème énonce que cette normalisation semble une bonne approximation sur la longueur d'un plus court vecteur du réseau. Il existe plusieurs preuve ; dans ce cours, on préfère une approche géométrique liée au comptage de points de réseaux dans des boules. Plus précisément, on va d'abord prouver le théorème suivant.

Théorème 1.16 (Théorème du corps convexe, Minkowski)**.** Soit $\mathscr{C}$ un convexe symétrique borné et $\mathscr{L}$ un réseau de rang n, tous deux dans $\mathbb{R}^n$. Si $\mathrm{Vol}(\mathscr{C}) > 2^n \det(\mathscr{L})$, alors $\mathscr{C}$ contient un vecteur non nul de $\mathscr{L}$.

Démonstration. On peut sans perte de généralité se ramener à $\mathscr{L} = \mathbb{Z}^n$. En effet, $\mathscr{L} = \mathbf{B}\mathbb{Z}^n$ pour une certaine transformation linéaire $\mathbf{B}$, et on a $\mathrm{Vol}(\mathbf{B}^{-1}\mathscr{C}) = \frac{\mathrm{Vol}(\mathscr{C})}{\det \mathscr{L}}$. Comme $\mathbf{B}^{-1}\mathscr{C}$ est aussi un convexe symétrique borné, il suffit donc de montrer que $\mathrm{Vol}(\mathscr{C}) > 2^n$, alors il existe un vecteur entier non nul dans $\mathscr{C}$.

On considère pour cela le "demi-convexe" $\mathscr{C}' = \{\mathbf{x}/2 \ : \ \mathbf{x} \in \mathscr{C}\}$. Notons que par hypothèse, $\mathrm{Vol}(\mathscr{C}') > 1$. On se propose de montrer qu'il existe nécessairement deux translatés distincts de $\mathscr{C}'$ par $\mathbb{Z}^n$ qui sont non disjoints. On raisonne par contradiction. Supposons qu'aucun de ces translatés ne s'intersectent. Pour $R > 0$, on considère la famille $\mathscr{F}_R = \{\mathscr{C}' + \mathbf{u} \ : \ \mathbf{u} \in ([-R,R] \cap \mathbb{Z})^n\}$. Notons D le diamêtre de $\mathscr{C}$, et considérons le cube $K = [-R-D, R+D]^n$, qui par construction contient toute la famille $\mathscr{F}_R$. On a donc $\mathrm{Vol}(\mathscr{F}_R) = (2R+1)^n \cdot \mathrm{Vol}(\mathscr{C}') \leqslant \mathrm{Vol}(K) = (2R+2D)^n$, ou autrement dit

$$\mathrm{Vol}(\mathscr{C}') \leqslant \left(1 + \frac{2D-1}{2R+1}\right)^d.$$

Mais comme R est abitraire, on en déduit que $\mathrm{Vol}(\mathscr{C}') \leqslant 1$, une contradiction.

Il existe donc deux translatés de $\mathscr{C}'$ non disjoints, ou de manière équivalente, un vecteur non nul $\mathbf{u} \in \mathbb{Z}^n$ tel que $\mathscr{C}' \cap \mathscr{C}' + \mathbf{u}$ est non vide. Ainsi, il existe $\mathbf{x} \in \mathscr{C}'$ tel que $\mathbf{x} - \mathbf{u} \in \mathscr{C}'$. Comme $\mathscr{C}'$ est symétrique, $\mathbf{u} - \mathbf{x}$ est aussi dans $\mathscr{C}'$, et par convexité, le segment $[\mathbf{x}, \mathbf{u} - \mathbf{x}]$ est inclu dans $\mathscr{C}'$. A fortiori, le milieu $\mathbf{m} = \mathbf{u}/2$ est aussi dans $\mathscr{C}'$, et le résultat en découle. □

La preuve du Théorème 1.15 est une conséquence rapide.

Preuve du Théorème 1.15. La boule ouverte $\mathscr{B}(\lambda_1(\mathscr{L}))$ est un convexe symétrique borné qui ne contient, par définition, aucun point non nul du réseau $\mathscr{L}$. D'après le théorème précédent, son volume est donc inférieur ou égal à $2^n \det(\mathscr{L})$. D'autre part, cette boule contient le cube centré en 0 et de côté $2\lambda_1(\mathscr{L})/\sqrt{n}$. On conclut en comparant les volumes décrits. □

Commentaires :

- On peut avoir une borne explicite un peu plus fine en utilisant le volume de la boule Euclidienne à la place du volume du cube inscrit, mais elle est moins esthétique.
- il est facile de constuire des réseaux ayant un plus court vecteur arbitrairement plus petit que cette borne (exercice).
- Cette borne supérieure est en fait optimale, à constante près. Plus précisément, pour tout n, il existe un réseau de rang n tel que $\lambda_1(\mathscr{L}) \leqslant c\sqrt{n}\det(\mathscr{L})^{1/n}$, pour une constante c.

Il est aussi naturel d'étudier la *constante d'Hermite*

$$\gamma_n = \sup_{\mathscr{L} \,:\, \mathrm{rk}\,\mathscr{L} = n} \frac{\lambda_1(\mathscr{L})^2}{\det(\mathscr{L})^{2/n}}.$$

Seules ses valeurs pour $n \leqslant 8$ sont connues ; par exemple, $\gamma_2 = \frac{4}{3}$, ce qui donne une borne sur $\lambda_1(\mathscr{L})$ légèrement meilleure que la borne du Théorème 1.15 en rang 2. On a cependant :

Théorème 1.17 (Hermite)**.** Pour tout $n \geqslant 2$, on a $\gamma_n \leqslant (\gamma_2)^{n-1}$.

Plus loin dans ce cours, on verra l'algorithme LLL de Lenstra, Lenstra et Lovàsz, qui peut être interprêté comme une version *effective* de cette inégalité.

2. Orthogonalisation de Gram-Schmidt

Rappel sur l'orthogonalité. On rappelle que dans une espace euclidien $(V, \langle , \rangle)$, deux vecteurs x, y sont dits orthogonaux lorsque $\langle x, y\rangle = 0$. Pour un sous-espace F de V, l'orthogonal de F est le sous-espace

$$F^\perp = \{y \in V \ : \ \langle x, y\rangle = 0 \ \forall \, x \in F\}.$$

On rappelle que $V = F \oplus F^\perp$, en particulier $\dim F^\perp = \dim V - \dim F$ et $F \cap F^\perp = \{0\}$. Une projection est une transformation linéaire π telle que $\pi^2 = \pi$. Elle est de plus dite orthogonale lorsque $\langle \pi(x), y\rangle = \langle x, \pi(y)\rangle$ pour tout $x, y \in V$. Dans ce cas, il existe une unique projection $\pi^\perp$ telle que $\mathrm{id} = \pi + \pi^\perp$; comme on s'y attend, si l'image de π est le sous-espace F, alors l'image de $\pi^\perp$ est le sous-espace $F^\perp$. De manière équivalente, on a dans ce cas $\ker \pi = F^\perp$ et $\ker \pi^\perp = F$.

Soit (b_i) une base du sous-espace F et $\mathbf{B}$ la matrice colonne de ces vecteurs dans une base fixée (e_i) de V (la base canonique par exemple). Alors la matrice de π dans la base des (e_i) est $\mathbf{P} = \mathbf{B}(\mathbf{B}^t\mathbf{B})^{-1}\mathbf{B}^t$, et celle de $\pi^\perp$ est $\mathbf{I}_n - \mathbf{P}$.

Une famille $(b_i)_i$ est dite orthogonale si $\langle b_i, b_j\rangle = 0$ pour tout $i \neq j$. On vérifie qu'une famille orthogonale est nécessairement libre. Si une famille orthogonale $(b_i)_i$ est de plus une base de V, on peut décrire les coordonnées d'un vecteur dans la base $(b_i)_i$ à l'aide des formes quadratiques/bilinéaires associées :

$$x = \sum_i \frac{\langle x, b_i\rangle}{\|b_i\|^2} b_i.$$

En particulier, si on note P_i la projection orthogonale sur le sous-espace $\mathbb{R}b_i$, on a $P_i(x) = \frac{\langle x, b_i\rangle}{\|b_i\|^2} b_i$, ou encore, la forme linéaire $x \mapsto \frac{\langle x, b_i\rangle}{\|b_i\|^2}$ donne la coordonnée de x sur b_i.

Exercice 2.1. Pour se dérouiller sur les formes bilinéaires et l'orthogonalité, remontrer toutes les propriétés rappelées ci-dessus. Montrer aussi le théorème de Pythagore : si x, y sont orthogonaux, on a $\|x+y\|^2 = \|x\|^2 + \|y\|^2$.

2.1. Le procédé de Gram-Schmidt.

Il est connu qu'il existe toujours des bases orthogonales pour les espaces euclidiens, mais mieux, le procédé d'orthogonalisation de Gram-Schmidt fournit un algorithme pour en calculer. Ce procédé est incontournable lorsqu'on cherche à manipuler algorithmiquement un réseau euclidien. Soit donc une famille libre $(b_i)_{1 \leqslant i \leqslant n}$ de $\mathbb{R}^m$. On définit une nouvelle famille $(b_i^*)_{1 \leqslant i \leqslant n}$ par le procédé suivant :

$$b_1^* = b_1, \quad \text{et pour tout } 2 \leqslant i \leqslant n, \ b_i^* = b_i - \sum_{j=1}^{i-1} \frac{\langle b_i, b_j^*\rangle}{\|b_j^*\|^2} b_j^*.$$

Intuitivement, on construit chaque nouveau vecteur en "enlevant orthogonalement" la contribution de l'espace vectoriel déjà engendré. Formellement, on a $b_2^* = b_2 - P_1(b_2) = P_1^\perp(b_2)$, $b_3^* = b_3 - P_1(b_3) - P_2(b_3) = b_3 - P_{\mathrm{span}(b_1, b_2)}(b_3) = P^\perp_{\mathrm{span}(b_1, b_2)}(b_3)$, etc. Pour ce cours, on préferera une preuve "en coordonnées" permettant d'introduire quelques notations utiles.

Proposition 2.2. La famille $(b_i^*)_{1 \leqslant i \leqslant n}$ est orthogonale et pour tout $1 \leqslant i \leqslant n$, on a $\mathrm{span}_{\mathbb{R}}(b_1, \dots b_i) = \mathrm{span}_{\mathbb{R}}(b_1^*, \dots, b_i^*)$.

Démonstration. La preuve se fait par récurrence. Pour $k = 1$ c'est clair. Supposons la proposition vraie pour jusqu'à $k - 1 \leqslant n - 1$, et montrons qu'elle est alors vraie pour k. Notons $\mu_{i,j} = \frac{\langle b_i, b_j^*\rangle}{\|b_j^*\|^2}$. Par définition, on peut écrire pour tout

$i < k-1$

$$\begin{aligned}\langle b_k^*, b_i^*\rangle &= \langle b_k, b_i^*\rangle - \sum_{j\leqslant k-1} \mu_{k,j}\langle b_j^*, b_i^*\rangle \\ &= \langle b_k, b_i^*\rangle - \mu_{k,i}\|b_i^*\|^2 \\ &= 0,\end{aligned}$$

où la deuxième ligne s'obtient par hypothèse de récurrence et la dernière par la définition de $\mu_{k,i}$. Ceci prouve l'orthogonalité des (b_i^*). Ensuite, par construction, b_k est dans l'espace engendré par les $(b_i^*)_{1\leqslant i\leqslant k}$, et b_k^* est dans l'espace engendré par $b_1^*, \dots, b_{k-1}^*$ et b_k. Ce second espace est par hypothèse de récurrence l'espace engendré par les $(b_i)_{1\leqslant i\leqslant k}$, ce qui conclut. □

Exercice 2.3. Faire la démonstration avec le formalisme des projections orthogonales (sans passer par des coordonnées).

Dans une base de $\mathbb{R}^m$, on peut écrire sous forme matricielle

$$\mathbf{B} = \mathbf{B}^*\mathbf{U}, \text{ avec } \mathbf{U} = \begin{bmatrix} 1 & \mu_{2,1} & \dots & & \mu_{n,1} \\ 0 & 1 & \dots & & \\ & & 1 & & \vdots \\ & & & 1 & \mu_{n,n-1} \\ & & & & 1 \end{bmatrix},$$

où $\mathbf{B}, \mathbf{B}^*$ sont les matrices colonnes des familles $(b_i), (b_i^*)$ respectivement. Les entrées $\mu_{i,j}$ de la matrice $\mathbf{U}$ seront appelées ici les *coordonnéees Gram-Schmidt* des b_i (parfois, les coefficients). Notons que $\det \mathbf{U} = 1$ mais que ses entrées sont a priori *des réels*. Il est toujours possible de normaliser itérativement les b_i^* pour se ramener à une base *orthonormée*. Cependant, ceci normalise aussi le volume à 1, faisant perdre une propriété du réseau sous-jacent : on évitera de le faire dans ce cours. D'ailleurs, l'orthogonalisation de Gram-Schmidt conserve l'information volumique du réseau engendré par les (b_i), et le résultat suivant est très intuitif : le volume d'un pavé est le produit des longeurs de ses côtés.

Proposition 2.4. Pour tout réseau $\mathscr{L} = \mathscr{L}(\mathbf{B})$ de rang n, on a $\det(\mathscr{L}) = \prod_{i=1}^n \|b_i^*\|$.

Démonstration. Il s'agit de "dépiler" les définitions : $(\det \mathscr{L})^2 = \det(\mathbf{B}^t\mathbf{B}) = \det(\mathbf{U})^2 \det(\mathbf{B}^{*t}\mathbf{B}^*)$, et de se rappeler que $\det \mathbf{U} = 1$ et que $\mathbf{B}^*$ a ses colonnes deux à deux orthogonales. □

Corollaire 2.5 (Inégalité de Hadamard)**.** Pour tout réseau de rang n, on a $\det \mathscr{L} \leqslant \prod_{i=1}^n \|b_i\|$.

Démonstration. On a $\|b_i\|^2 = \|b_i^*\|^2 + \sum_{j\leqslant i} \mu_{i,j}^2 \|b_i^*\|^2$, d'après le théorème de Pythagore. On en tire que $\|b_i\| \geqslant \|b_i^*\|$, puis le résultat. □

Exercice 2.6. Proposer une preuve sans passer par les $\mu_{i,j}$. Indication : considérer la norme d'une projection orthogonale.

On donne un dernier résultat étonamment utile, qui traduit la forme triangulaire de la matrice de μ_{ij}.

Lemme 2.7. Soit $b_1, \dots, b_n$ une base d'un réseau $\mathscr{L}$. Alors on a $\lambda_1(\mathscr{L}) \geqslant \min_i \|b_i^*\|$.

Démonstration. Soit $b \in \mathscr{L} \setminus \{0\}$. Il existe un plus grand entier $i_0 \leqslant n$ et des $x_i \in \mathbb{Z}$ avec $x_{i_0} \neq 0$ tels que $b = \sum_{i \leqslant i_0} x_i b_i$ (autrement dit, i_0 est le dernier indice où b a une coordonnée non nulle dans la base (b_i)). En utilisant la définition des Gram-Schmidt, on trouve certainement des rationnels x_i' tels que $b = x_{i_0} b_{i_0}^* + \sum_{i<i_0} x_i' b_i^*$. Le théorème de Pythagore donne alors $\|b\|^2 = |x_{i_0}|^2 \|b_{i_0}^*\|^2 + \sum_{i<i_0} |x_i'|^2 \|b_i^*\|^2$. Comme la somme dans le terme de droite est positive et que $x_{i_0} \in \mathbb{Z}$, il vient $\|b\|^2 \geqslant \|b_{i_0}^*\|^2 \geqslant \min_i \|b_i^*\|^2$. On obtient le résultat annoncé en choisissant un vecteur le plus court pour b. □

3. Réduction des bases des réseaux euclidiens, et LLL

3.1. **Principe de la réduction des réseaux euclidiens.** On a vu que toutes les bases d'un réseau diffèrent d'une transformation entière de déterminant ± 1. L'ensemble des ces transformations est aussi connu comme le groupe *unimodulaire*, et noté $\mathrm{GL}_n(\mathbb{Z})$. [3] On peut alors résumer le résultat ci-dessus par l'action de groupe

$$\begin{array}{rcl} \mathrm{GL}_n(\mathbb{Z}) \times \mathrm{GL}_n(\mathbb{R}) & \longrightarrow & \mathrm{GL}_n(\mathbb{R}) \\ (\mathbf{U}, \mathbf{B}) & \longmapsto & \mathbf{BU}, \end{array}$$

et un réseau correspond alors à une orbite de cette action. Sous ce point de vue, la *réduction de réseaux* consiste alors à trouver des "bons" représentants pour chaque orbite, où le terme "bons" dépend du contexte. En algorithmique et en cryptographie, un bon représentant est une base la plus orthogonale possible et impliquant les vecteurs les plus courts possibles. On aimerait aussi en trouver *constructivement et efficacement*. L'état de l'art suggère que c'est un problème difficile, et il faut faire un compromis entre la *qualité* (la longueur des vecteurs de la base) garantie par l'algorithme et le temps d'exécution.

3.2. **Première intuition et bases "size-réduites".** Il est possible de se donner une idée intuitive des mécanismes de réduction de réseaux en dimension 2. Considérons [4] un réseau donné par "une très mauvaise base" (b_1, b_2), où sans perte de généralité, on peut supposer que $\|b_1\| \leqslant \|b_2\|$: les vecteurs sont très longs, et $\langle \frac{b_1}{\|b_1\|}, \frac{b_2}{\|b_2\|} \rangle = \cos(b_1, b_2)$ est assez proche de ± 1. Notons (b_1^*, b_2^*) la Gram-Schmidt, pour voir que $\|b_2^*\|^2 = \|b_2\|^2(1 - \cos(b_1, b_2)^2)$. Autrement dit, b_2^* est assez court. Ce n'est pas un vecteur du réseau, mais il y en a pas loin : avec les notations de la section précédente, considérer le vecteur $b_2' = b_2 - \lfloor \mu_{2,1} \rceil b_1$.

On a donc une nouvelle base du réseau, et il est possible que b_2' soit vraiment plus court que b_1 : dans ce cas on a *progressé*, et on peut recommencer le procédé sur la base (b_2', b_1). Chaque fois qu'on réduit de cette manière les vecteurs, le produit $\|b_1\| \cdot \|b_2\|$ diminue. On sait que le produit ne peut pas se réduire éternellement car un réseau possède toujours un plus court vecteur ; le problème est qu'on n'a aucune idée de la valeur de λ_1 *a priori*. Par contre, les transformations effectuées préservent le réseau et donc son volume : l'inégalité de Hadamard nous donne donc une garantie effective. En admettant qu'on gagne un facteur plus grand que 1 sur le produit à chaque étape, un tel algorithme a de bonne chances de se terminer rapidement. Ceci décrit informellement l'algorithme de Gauss-Lagrange, et on verra en TD qu'il permet d'obtenir des plus courts vecteurs en dimension 2.

3. Cette notation se justifie car ces matrices correspondent exactement aux matrices entières dont l'inverse est aussi une matrice entière.

4. Et un jour, il y aura des dessins, si si ! Ce serait bien parce que c'est vachement visuel en fait.

Algorithm 1: Size-réduction

Entrées: une base $b_1,\dots,b_n$ d'un réseau.

Sortie: une base $b_1,\dots,b_n$ size-réduite.

Calculer $b_1^*,\dots,b_n^*$ la Gram-Schmidt des $(b_i)_i$;

pour $i=2$ *jusqu'à* n **faire**

  pour $j=i-1$ *jusqu'à* 1 **faire**

    $b_i \leftarrow b_i - \left\lfloor \frac{\langle b_i, b_j^*\rangle}{\|b_j^*\|^2} \right\rceil b_j$;

  fin pour

fin pour

Renvoyer $b_1, b_2, \dots, b_n$

Exercice 3.1. Dans la procédure ci-dessus, montrer que si $|\mu_{2,1}|>1$, alors $\|b_2\|^2 \leqslant 3\|b_2'\|^2$.

En filligrane dans la procédure ci-dessus, on a vu l'idée fondamentale de prendre comme nouvelle base la meilleure approximation entière des Gram-Schmidt de la base de départ. On formalise maintenant cette notion.

Définition 3.2. Soit $b_1,\dots,b_n$ une base d'un réseau, et $\mathbf{U}$ la matrice triangulaire supérieure telle que $\mathbf{B}=\widetilde{\mathbf{B}}\mathbf{U}$ est dite *size-réduite* [5] si $\max_{1\leqslant i<j\leqslant n}|\mu_{i,j}| \leqslant \frac{1}{2}$.

Les idées présentées donnent aussi lieu à l'Algorithme 1 pour obtenir une base size-réduite.

Proposition 3.3. L'algorithme 1 renvoie une base size-réduite de $\mathscr{L}(b_1,\dots,b_n)$.

Démonstration. La propriété qui fait "marcher la preuve" est que les boucles ne modifient pas les Gram-Schmidt (b_i^*). Notons π_{i-1} la projection orthogonale sur $\mathrm{span}_{\mathbb{R}}(b_1,\dots,b_{i-1})^{\perp}$, pour que par définition, $b_i^*=\pi_{i-1}(b_i)$. Comme $\mathrm{span}(b_1,\dots,b_{i-1})=\mathrm{span}(b_1^*,\dots,b_{i-1}^*)$ par la Proposition 2.2, on a de plus pour tout $1\leqslant j\leqslant i-1$ et tout $\alpha\in\mathbb{R}$ que $\pi_{i-1}(b_i-\alpha b_j)=\pi_{i-1}(b_i)$, ce qui implique l'invariance de Gram-Schmidt tout au long de l'algorithme. Notons maintenant $b_i^{(j)}$ le vecteur obtenu après la soustraction d'un multiple de b_j à b_i dans la deuxième boucle. Pour $1\leqslant j< i-1$, on a par définition

$$\langle b_i^{(j)}, b_j^*\rangle = \langle b_i^{(j+1)}, b_j^*\rangle - \left\lfloor \frac{\langle b_i^{(j+1)}, b_j^*\rangle}{\|b_j^*\|^2}\right\rceil \langle b_j, b_j^*\rangle.$$

Bien qu'à priori, les b_i actuels ne sont plus ceux de départ, l'invariance des Gram-Schmidt nous donne $\langle b_j, b_j^*\rangle = \|b_j^*\|^2$. Ainsi, la j-ème coordonnée de Gram-Schmidt de $b_i^{(j)}$ est $\mu_{i,j}' := \frac{\langle b_i^{(j)}, b_j^*\rangle}{\|b_j\|^2}$, et l'égalité ci-dessus nous donne $|\mu_{i,j}'|\leqslant 1/2$. On remarque ensuite, par exemple en regardant la matrice $\mathbf{U}$, que retrancher un multiple de b_j à un vecteur n'agit que sur ses j premières coordonnées dans la base des Gram-Schmidt. Les étapes suivantes de la deuxième boucle ne modifient donc pas [6] les $\mu_{i,j}$ déjà réduits. Il reste à montrer que le réseau est préservé, mais il n'est pas difficile de se

5. Désolé pour les anglicismes.

6. C'est pour ça qu'on part de la fin dans la deuxième boucle.

convaincre que la matrice correspondant à un passage dans la boucle interne est entière de déterminant 1 : en fait, elle [7] s'écrit $\mathrm{Id}_n - \lfloor \frac{\langle b_i, b_j^* \rangle}{\|b_j^*\|^2} \rceil \delta_{j,i}$. □

3.3. **L'algorithme LLL.** Informellement, l'idée derrière l'algorithme de Gauss-Lagrange est d'enchainer des size-réductions, quitte à parfois échanger b_1 et b_2 lorsque $\|b_2\|$ est sensiblement plus court que $\|b_1\|$, c'est-à-dire qu'on fait un certain "progrès" dans notre réduction de la base de départ. Il est naturel de chercher à étendre cette approche en plus grande dimension, mais la situation est alors plus problématique. Il faut pouvoir donner un critère quantitatif pour mesurer ce progrès, afin de savoir aussi quand on a besoin de permuter des vecteurs. D'autre part, il faut que ce progrès permette de gagner un facteur constant sur l'inégalité de Hadamard à chaque étape pour espérer n'avoir besoin que d'un nombre polynomial de size-réduction.

Ces idées ont conduit Lenstra, Lenstra et Lovàsz à l'algorithme LLL en 1982. Ses applications sont nombreuses, allant de la factorisation de polynômes rationnels à la théorie algébrique des nombres, en passant par la cryptanalyse [8]. Il peut aussi fournir une approche *effective* au Théorème 1.17. Dans ce cours, on fera que présenter l'algorithme [9]. On commence d'abord par la notion de base LLL-réduite.

Définition 3.4. Une base $b_1, \dots, b_n$ d'un réseau est dite LLL-réduite si

— elle est size-réduite ;

— **Condition de Lovàsz :** pour tout $1 \leqslant i < n$, on a $\frac{3}{4}\|b_i^*\|^2 \leqslant \|\mu_{i+1,i} b_i^* + b_{i+1}^*\|^2$.

L'intérêt des bases [10] LLL-réduites est résumé par la proposition suivante : les Gram-Schmidt ne décroissent pas trop vite, et le premier vecteur de la base ne peut pas être trop long.

Proposition 3.5. Soit $b_1, \dots, b_n$ une base d'un réseau $\mathscr{L}$. On a

— $\|b_i^*\|^2 \leqslant 2\|b_{i+1}^*\|^2$ pour $1 \leqslant i < n$;

— $\|b_1\| \leqslant 2^{(n-1)/2} \lambda_1(\mathscr{L})$.

Démonstration. Combiner la propriété de size-réduction et la condition de Lovàsz avec le théorème de Pythagore implique le premier point. Pour le second, on note qu'utiliser plusieurs fois la première propriété donne $\|b_1\| \leqslant 2^{(i-1)/2}\|b_i^*\|$ pour tout $1 < i \leqslant n$, et donc a fortiori $\|b_1\| \leqslant 2^{(n-1)/2} \min_i \|b_i^*\|$. On conclut avec le Lemme 2.7. □

7. On les appelle aussi des matrices de transvection.
8. Bien sûr, c'est utile pour des schémas fondés sur les réseaux comme des problème de type sac-à-dos, mais aussi dans des attaques sur RSA ou des courbes elliptiques !
9. Une analyse donnant une complexité polynomiale sera proposée en annexe, un jour.
10. On peut généralement remplacer $3/4$ par un paramètre $\frac{1}{4} < \delta < 1$, ce qui permet parfois d'affiner certaines estimations.

A contrario, si la condition de Lovàsz n'est pas satisfaite, on peut certainement progresser dans la réduction en échangeant b_i et b_{i+1}. Ceci conduit à la formulation suivante pour LLL.

Algorithm 2: LLL

Entrées: une base $b_1, \dots, b_n$ d'un réseau.

Sortie: une base $b_1, \dots, b_n$ LLL-réduite.

Etape 1 : Calculer $b_1^*, \dots, b_n^*$ la Gram-Schmidt des $(b_i)_i$;

Size-réduire $(b_1, \dots, b_n)$ avec l'algorithme 1;

Etape 2 : si *il existe* $1 \leqslant i \leqslant n-1$ *tel que* $\frac{3}{4}\|b_i^*\|^2 > \|\mu_{i+1,i}b_i^* + b_{i+1}^*\|^2$ **alors**

  Echanger b_i et b_{i+1};

  Revenir à l'étape 1;

fin si

Renvoyer $b_1, \dots, b_n$

Théorème 3.6. L'algorithme 2 se termine en un nombre polynomial d'étapes et manipule des nombres dont la taille en bits est polynomiale en la taille de la base en entrée.
(NGuyen-Stehlé, 2002) Il existe une variante de complexité $O(n^{4+\varepsilon} \log \|\mathbf{B}\|_\infty (n + \log \|\mathbf{B}\|_\infty))$, où $\|\mathbf{B}\|_\infty$ est la valeur absolue de la plus grande coordonnées des b_i.

Il est clair que si l'algorithme se termine, il produit une base du réseau qui est LLL-réduite. Si on ne se préoccupe pas de la taille des nombres manipulés, on peut montrer qu'on effectue un nombre polynomial d'échange avec le même argument que dans l'algorithme de Gauss-Lagrange : le produits des $\|b_i\|^2$ diminue à chaque échange d'un facteur constant ($\frac{3}{4}$), et on ne peut pas descendre sous le volume du réseau (informellement).

4. Quelques problèmes algorithmiques de réseaux euclidiens

4.1. Chercher des vecteurs courts.

Le *calcul de (plus) courts vecteurs* dans des réseaux est un problème notoirement difficile, qui sert de fondation de sécurité à de nombreuses primitives cryptographiques dite "post-quantiques". Plus précisément, considérons le problème suivant, paramétré par le rang n du réseau :

— **Svp** *(pour Shortest Vector Problem) : étant donnée une base* $\mathbf{B}$ *d'un réseau* $\mathscr{L}$*, trouver* $\mathbf{v} \neq 0$ *tel que* $\|\mathbf{v}\| = \lambda_1(\mathscr{L})$.

Ajtai a montré que ce problème était NP-complet. En l'état des connaissances, et pour des réseaux arbitraires, on ne connait que des algorithmes demandant un nombre au moins exponentiel d'opérations pour résoudre ce problème, et ce même en s'autorisant des algorithmes *quantiques* [11]. Citons les principaux :

(1) les algorithmes de type énumeration. Comme le nom l'indique, il s'agit d'énumérer tous les vecteurs du réseau qui sont dans une certaine boule bien choisie. Le temps d'exécution de ces algorithmes est généralement en $2^{O(n^2)}$, mais en pratique, ils sont très utilisés pour des dimensions allant jusqu'à $n \approx 80$ (avec de nombreuses optimisations et sous certains arguments heuristiques);

11. D'autres problèmes tels que la factorisation ou le problème du logarithme discret, fondamentaux dans les protocoles de cryptographie à clé publique actuellement utilisés, sont "cassés" par des algorithmes quantiques : on connait des algorithmes quantiques polynomiaux pour les résoudre.

(2) les algorithmes de type crible. Le principe consiste à générer deux listes d'éléments du réseau, puis de construire la liste de toutes les différences entre les éléments des deux listes. On espère obtenir des vecteurs plus courts, puis on recommence le procédé. Pour prouver qu'on peut ainsi diminuer la taille des vecteurs, les listes doivent être de taille exponentielle en la dimension. On a donc besoin d'une grande quantité de mémoire pour stocker les listes nécessaires. Le temps d'exécution est ensuite en [12] $2^{O(n)}$.

En cryptographie, on préfère de loin se reposer sur la variante *relaxée* du problème :

— $\mathbf{SVP}_\gamma$, où $\gamma > 0$: *étant donnée une base* $\mathbf{B}$ *d'un réseau* $\mathscr{L}$*, trouver* $\mathbf{v} \neq 0$ *tel que* $\|\mathbf{v}\| \leqslant \gamma \cdot \lambda_1(\mathscr{L})$.

Le paramètre γ ici doit être pensé comme une fonction du rang n. Parfois, on appelera un vecteur solution de $\mathbf{SVP}_\gamma$ un *vecteur court*. L'état des connaissances peut se résumer informellement de la manière suivante :

— $\gamma = O(1)$ reste NP-complet.

— Lorsque $\gamma = \text{poly}(n)$, on ne connait que des algorithmes (quantiques, classiques) demandant un temps d'exécution exponentiel. La sécurité de schémas cryptographiques repose sur ce régime de paramètres.

— Lorsque $\gamma = 2^{O(n)}$, l'algorithme LLL résout le problème en temps polynomial, d'après la Proposition 3.5.

Pour la culture : les régimes intermédiaires entre $\text{poly}(n)$ et $2^{O(n)}$ peuvent être atteint par des familles d'algorithmes, dont la complexité évolue "inversement" avec le facteur d'approximation. Enfin, même le problème de *déterminer* λ_1 est difficile, même en s'autorisant [13] une erreur γ, et même s'il est a priori moins difficile que de celui calculer un court vecteur .

4.2. **Chercher des vecteurs proches.** Les problèmes de décodages sont classiques en informatique et théorie du signal. Il en existe plusieurs pour les réseaux euclidiens, mais on se contentera de présenter le problème suivant (et une variante) :

— **CVP** : étant donnés une cible $t \in \mathbb{R}^m$ et un réseau $\mathscr{L}(\mathbf{B})$, trouver $v \in \mathscr{L}$ tel que $\|t-v\| = d(t,\mathscr{L}) := \min\{\|t-v\| \ : \ v \in \mathscr{L}\}$.

— $\mathbf{CVP}_\gamma$: pour $\gamma \geqslant 1$, étant donnés une cible $t \in \mathbb{R}^m$ et un réseau $\mathscr{L}(\mathbf{B})$, trouver $v \in \mathscr{L}$ tel que $\|t-v\| = \gamma \cdot d(t,\mathscr{L})$.

Le problème **CVP** est difficile pour un réseau arbitraire, bien que dans certaines familles de réseaux très spécifiques comme $\mathbb{Z}^n$, on connaisse des algorithmes polynomiaux (parfois même quasi-linéaires [CS88]). La qualité de la base dans laquelle on effectue le décodage est prépondérante, et on ne sait pas prouver bien mieux que le résultat ci-dessous en général.

Théorème 4.1. Il existe un algorithme qui résout $\mathbf{CVP}_{\exp(n)}$ en temps polynomial.

Eléments de preuve. Il faut "une bonne base" en temps polynomial, donc on commence par lancer LLL sur la base donnée par l'instance. Intuitivement, les vecteurs renvoyés sont au pire "exponentiellement" plus grands que les plus courts possibles. Si on utilise ensuite l'algorithme *Nearest Plane* de Babai (voir section suivante et l'Algorithme 4), il est possible de transférer cette approximation exponentielle sur la base et la décroissance des Gram-Schmidt en une approximation sur la distance entre le vecteur renvoyé par l'algorithme et la cible — voir le TD 2. □

12. Les constantes dans le "grand O" sont sujettes à certains débats dans la communauté.

13. Le théorème de Minkowski (Théorème 1.15) rend le problème trivial si on s'autorise $\gamma \geqslant \sqrt{n}$.

5. Les algorithmes de Babai

5.1. **"Vecteur" le plus proche d'une cible, en dimension 1.** D'après l'Exercice 1.5, les réseaux euclidiens de $\mathbb{R}$ sont de la forme $\alpha\mathbb{Z}$, et on peut supposer $\alpha \geqslant 0$. Soit maintenant $t \in \mathbb{R}$ arbitraire. Le point de $\alpha\mathbb{Z}$ le plus proche [14] de t est le multiple entier de α le plus proche de t, c'est-à-dire $\alpha \cdot \lfloor \frac{t}{\alpha} \rceil$. Résoudre le problème **CVP** en dimension 1 est donc très facile.

5.2. **Algorithme "Round-off".** Moralement, l'approche du paragraphe précédent a consisté à arrondir la coordonnée de t selon le "vecteur" α à l'entier le plus proche. Il est très naturel d'étendre cette méthode au cas général, ce qui conduit à l'Algorithme 3. Pour simplifier, on considère le cas d'un réseau de rang plein ; il n'est pas difficile de l'étendre au cas général en se ramenant à la projection orthogonale de la cible sur l'espace engendré par le réseau. On rappelle que $\mathscr{D}(\mathbf{B})$ désigne le domaine fondamental de la base $\mathbf{B}$.

Algorithm 3: Round-off de Babai

Entrées: une base $\mathbf{B} = [b_1, \dots, b_n]$ d'un réseau $\mathscr{L}$, une cible $t \in \mathbb{R}^n$.

Sortie: un élément $v \in \mathscr{L}$ tel que $t - v \in \mathscr{D}(\mathbf{B})$.

$\tilde{z} \leftarrow \mathbf{B}^{-1} t$;

$z \leftarrow \lfloor \tilde{z} \rceil := (\lfloor \tilde{z}_1 \rceil, \dots, \lfloor \tilde{z}_n \rceil)$;

Renvoyer $v = \mathbf{B}z$;

Proposition 5.1. L'Algorithme 3 est correct et on a $\|t - v\| \leqslant \frac{n}{2} \cdot \max_i \|b_i\|$.

Démonstration. Le vecteur z construit est clairement entier, donc $v \in \mathscr{L}$. Par construction, on a $t - v = \mathbf{B}(\tilde{z} - z) = \sum_i (\tilde{z}_i - z_i) b_i$ avec $\tilde{z}_i - z_i \in (-\frac{1}{2}, \frac{1}{2}]$. Enfin, l'inégalité triangulaire donne $\|t - v\| \leqslant \sum_i |\tilde{z}_i - z_i| \cdot \|b_i\|$ et on conclut. □

Remarque 5.2. On peut aussi bien conclure que $\|t - v\| \leqslant \frac{\sqrt{n}}{2} \|\mathbf{B}\|_2$, où $\|\mathbf{B}\|_2 = \max \frac{\|\mathbf{B}x\|}{\|x\|}$ est la norme d'opérateur induite par la norme euclidienne, et est égale à la plus grande *valeur singulière* de $\mathbf{B}$. Bien que cette borne soit généralement plus fine, il est généralement plus difficile d'estimer les valeurs singulières "à la main" même quand les longueurs des colonnes sont connues.

On constate rapidement que si on n'a aucune garantie sur la base $(b_i)_i$, la solution renvoyée par l'Algorithme 3 peut être très mauvaise selon la cible. Par exemple, même en dimension 2, si $\mathscr{D}(\mathbf{B})$ est très écrasée, on peut facilement construire des cibles pour lesquelles l'algorithme renvoie 0, mais qui sont pourtant clairement plus proche de b_1 ou b_2.

Exercice 5.3. Construire un exemple comme décrit ci-dessus.

5.3. **Algorithme "Nearest Plane".** On présente maintenant un algorithme en général meilleur pour résoudre **CVP**. Informellement, son principe consiste à se rappeler qu'en dimension 1, on sait très bien résoudre le problème. On essaie donc de se ramener récursivement à cette dimension, et on espère que dépiler la récursion conserve un peu de précision.

14. Si t est un multiple entier de $\alpha/2$, il y a bien entendu deux solutions.

Géométriquement, diminuer la dimension de 1 consiste à trouver l'hyperplan affine le plus proche possible de la cible et contenant un point du réseau. Comme cet hyperplan est le plus proche, il y a de bonne chance pour que le point du réseau le plus proche de la cible y soit aussi. On projette donc notre cible sur l'hyperplan, on se ramène à un problème vectoriel par translation, puis on recommence sur la cible projeté jusqu'à arriver en dimension 1. Grâce à la Gram-Schmidt de la base donnée en entrée, on peut facilement décrire des hyperplans et effectuer ces projections — voir l'exercice plus bas, et aussi la preuve de la Proposition 6.21 — et on obtient l'Algorithme 4 ci-dessous. Il est donné en version itérative et dans le cas d'un réseau de rang plein, pour plus de simplicité.

Algorithm 4: Nearest Plane de Babai

Entrées: une base $\mathbf{B} = [b_1, \dots, b_n]$ d'un réseau $\mathscr{L}$, une cible $t \in \mathbb{R}^n$.

Sortie: un élément $v \in \mathscr{L}$ tel que $t - v \in \mathscr{D}(\widetilde{\mathbf{B}})$.

Calculer $\widetilde{\mathbf{B}} = [\widetilde{b_1}, \dots, \widetilde{b_n}]$ la Gram-Schmidt de $\mathbf{B}$;

$v \leftarrow 0, c \leftarrow t$;

pour $i = n$ *à* 1 **faire**

  $\tilde{c} \leftarrow \lfloor \frac{\langle c, \widetilde{b_i} \rangle}{\|\widetilde{b_i}\|^2} \rceil$; `/* l'hyperplan le plus proche de` c `est` $\tilde{c}\widetilde{b_i} + \mathrm{span}_{\mathbb{R}}(b_1, \dots, b_{i-1})$ `*/`

  $v \leftarrow v + \tilde{c} b_i$;

  $c \leftarrow c - \tilde{c} b_i$;

fin pour

Renvoyer v;

Proposition 5.4. L'Algorithme 4 est correct et on a $\|t - v\|^2 \leqslant \frac{1}{4} \sum_i \|\widetilde{b_i}\|^2 \leqslant \frac{n}{4} \max_i \|\widetilde{b_i}\|^2$.

Démonstration. Par construction, v est dans $\mathscr{L}$. Si on admet un instant que $t - v \in \mathscr{D}(\widetilde{\mathbf{B}})$, alors la borne sur la distance vient du théorème de Pythagore. Notons avec un indice i les éléments $\tilde{c}, v, c$ calculés dans la boucle à l'indice i, et $t = \sum_j t_j \widetilde{b_j}$. Pour démontrer l'appartenance au domaine fondamental, on va montrer qu'à chaque étape, les dernières coordonnées de $t - v_i$ dans la base $(\widetilde{b_i})_i$ sont inférieure à $\frac{1}{2}$ en valeur absolue. Par définition (voir la Section 2.1), on a $b_n = \widetilde{b_n} + \sum_{j<n} \mu_{n,j} \widetilde{b_j}$, si bien qu'au premier passage, on a $t - v_n = (t_n - \tilde{c}_n)\widetilde{b_n} + \tilde{c}_n \sum_{j<n} \mu n, j \widetilde{b_j} + \sum_{j<n} t_j \widetilde{b_j}$. Ceci nous donne

$$\left| \frac{\langle t - v_n, \widetilde{b_n} \rangle}{\|\widetilde{b_n}\|^2} \right| = |t_n - \tilde{c}_n| \leqslant \frac{1}{2}.$$

Supposons maintenant que c'est vrai pour tout $i+1 \leqslant j \leqslant n$, et montrons que cela reste vrai pour l'indice i. À cette étape, on a $t - v_i = t - v_{i+1} - \tilde{c}_i b_i$. Par construction des Gram-Schmidt, on a $\langle b_i, \widetilde{b_j} \rangle = 0$ pour $j > i$, et pour ces mêmes j on a donc $\langle t - v_i, \widetilde{b_j} \rangle = \langle t - v_{i+1}, \widetilde{b_j} \rangle$. Pour $j = i$, on trouve

$$\langle t - v_i, \widetilde{b_i} \rangle = \langle t - v_{i+1}, \widetilde{b_i} \rangle - \tilde{c}_i \|\widetilde{b_i}\|^2,$$

et on conclut en se rappelant la définition de $\tilde{c}_i$. □

Remarque 5.5. L'inégalité sur la longueur de $t - v$ se réécrit ici comme $\|t - v\| \leqslant \frac{1}{2} \|\widetilde{\mathbf{B}}\|_F$, où $\|\widetilde{\mathbf{B}}\|_F^2 = \sum_{i,j} \widetilde{b}_{ij}^2$ est la norme de Frobénius de $\widetilde{\mathbf{B}}$.

Un problème est que même si on choisit l'hyperplan le plus proche, il est tout à fait possible que celui-ci ne contienne pas le vecteur de $\mathscr{L}$ le plus proche de la cible courante, et encore une fois, c'est possible même en dimension 2. Ces erreurs d'hyperplans s'accumulent en grande dimension, et c'est la raison intuitive pour laquelle on n'arrive pas à prouver une meilleure approximation que celle donnée par la qualité de réduction de la base au départ — et en temps polynomial, on n'a "que" LLL. Enfin on peut montrer *qu'en moyenne* l'Algorithme 4 est meilleur que l'Algorithme 3, mais on peut construire des exemples où le deuxième donne une meilleure solution que le premier.

Exercice 5.6. Construire un exemple en dimension 2 où Nearest Plane "se trompe" d'hyperplan.

6. Gaussiennes et réseaux euclidiens

La cryptographie fondée sur les réseaux euclidiens fait un usage intensif de (pseudo-)aléas, et particulièrement de distribution de type Gaussiennes. Il peut s'agir d'outils pour des preuves de sécurité, ou même d'algorithmes *d'échantillonnage* impliqués dans des cryptosystèmes. Ainsi, dans les signatures "à la GPV", une base publique sert à vérifier que la signature correspond bien à un point du réseau, et la base secrète, composée de vecteurs courts, à trouver des vecteurs proches d'une cible correspdonant au message à signer. Les distributions Gaussiennes permettent de générer des signatures *sans fuiter d'informations* sur la base du réseau utilisée (voir aussi la Section 7).

Généralement, les fonctions Gaussiennes apparaissent dans de nombreux domaines : en théorie du signal, en physique, en statistiques, etc. Ce qui les rend attirantes en cryptographie, c'est tout d'abord leur très bons comportements sous transformations linéaires [15], mais aussi en convolution. En effet, ce sont des distributions avec lesquelles on peut souvent calculer *explicitement* des quantités importantes pour la sécurité, car leur concentration est très bien comprise en plus de leur bonnes propriétés algébriques. Dans cette section, on verra qu'elles interviennent aussi par leur propriétés harmoniques agréables, ce qu'on résume souvant en disant qu'elles sont des fonctions propres pour la transformée de Fourier. En particulier, on va avoir besoin de quelques rappels d'analyse harmonique et de dualité des réseaux euclidiens pour introduire convenablement l'objet central de cette section : les distributions Gaussiennes *discrètes*.

6.1. **Réseau dual.**

Définition 6.1. Le dual d'un réseau $\mathscr{L} \subset \mathbb{R}^m$ est $\mathscr{L}^\vee = \{x \in \mathbb{R}^m \ : \ \forall y \in \mathscr{L}, \langle x, y\rangle \in \mathbb{Z}\}$.

On rappelle que si $(b_i)_i$ est une base de $\mathbb{R}^m$, on définit sa base duale $(b_i^\vee)_i$ par les relations suivantes :

$$\langle b_i^\vee, b_j\rangle = \begin{cases} 1 \text{ si } i = j \\ 0 \text{ sinon.} \end{cases}$$

Sous forme matricielle, si on appelle $\mathbf{D}$ la base duale de $\mathbf{B}$, on constate que $\mathbf{D} = \mathbf{B}^{-t}$ lorsque le réseau est de rang plein. Si $\mathscr{L}$ n'est pas de rang plein, on a $\mathbf{D} = \mathbf{B}(\mathbf{B}^t\mathbf{B})^{-1}$. Bien sûr, la duale de la base duale est la base de départ.

Lemme 6.2. On a $\mathscr{L} = \mathscr{L}(b_1, \dots, b_m)$ si et seulement si $\mathscr{L}^\vee = \mathscr{L}(b_i^\vee, \dots, b_n^\vee)$. Ainsi, $\mathscr{L}^\vee$ est un réseau de même rang que $\mathscr{L}$ et $\det \mathscr{L}^\vee = (\det \mathscr{L})^{-1}$.

15. C'est même parfois une façon de les définir en grande dimension

Démonstration. Par définition de la base duale, on a clairement $\mathscr{L}(b_1^\vee,\dots,b_m^\vee)\subset\mathscr{L}^\vee$. Pour l'autre inclusion, soit $x\in\mathscr{L}^\vee$, et écrivons $x=\sum_i x_i b_i^\vee$ pour des x_i a priori réels. On a alors $\langle x,b_i\rangle = x_i\in\mathbb{Z}$ et l'inclusion voulue, puis le rang. L'identité entre les déterminants est immédiate en regardant les versions matricielles des bases. □

Lemme 6.3. Pout tout $a\in\mathbb{R}^*$, on a $(a\mathscr{L})^\vee=\frac{1}{a}\mathscr{L}^\vee$. Si $\mathscr{L}=\mathbb{Z}u$ pour $u\in\mathbb{R}^m$, alors $\mathscr{L}^\vee=\frac{1}{\|u\|^2}\mathbb{Z}u$.

Exercice 6.4. Montrer le Lemme 6.3. Quel est le dual de $\mathbb{Z}^n$? Quel est le dual de $\Lambda_q^\perp(\mathbf{A})$? De $\Lambda_q(\mathbf{A})$? Montrer que $(\mathscr{L}_1\oplus\mathscr{L}_2)^\vee=\mathscr{L}_1^\vee\oplus\mathscr{L}_2^\vee$.

6.2. **Rappels d'analyse harmonique.** Il n'est pas dans l'objectif du cours d'utiliser trop d'analyse harmonique abstraite. En particulier, comme il est d'ailleurs de coutume en cryptographie fondée sur les réseaux euclidiens, on ne cherchera pas les énoncés les plus précis, ni à rentrer dans les détails analytiques des propriétés utilisées. D'ailleurs, ils seront la plupart du temps utilisés avec des fonctions Gaussiennes, pour lesquelles "tout marche bien".

On supposera toujours $\mathbb{R}^m$ muni de sa mesure de Lebèsgue usuelle. Si f est une fonction intégrable sur $\mathbb{R}^m$, on définit sa *transformée de Fourier* par :

$$\hat{f}(\xi)=\int_{\mathbb{R}^m} f(x)e^{-2i\pi\langle x,\xi\rangle}dx.$$

En physique, la propriété suivante dit qu'un "décalage dans le temps" équivaut à une "rotation dans l'espace des phases".

Lemme 6.5. Soit $\tau_c(x)=x+c$. Si f est intégrable sur $\mathbb{R}^m$, alors $\widehat{f\circ\tau_c}(\xi)=\hat{f}(\xi)e^{2i\pi\langle\xi,c\rangle}$.

Démonstration. On dépile doucement les définitions :

$$\widehat{f\circ\tau_c}(\xi)=\int_{\mathbb{R}^m} f(x+c)e^{-2i\pi\langle\xi,x\rangle}dx=\int_{\mathbb{R}^m} f(x)e^{-2i\pi\langle\xi,x-c\rangle}dx=e^{2i\pi\langle\xi,c\rangle}\hat{f}(\xi).$$

□

La célèbre formule suivante est due à Poisson, et relie élégamment une fonction prise sur un réseau à sa transformée de Fourier sur le dual. Elle s'applique à des fonctions à décroissances assez rapides, que nous appellerons abusivement dans ce cours "des fonctions raisonnables". Les Gaussiennes sont le prototype de fonctions à décroissance rapide, donc ce n'est pas un abus abusif.

Théorème 6.6 (Formule sommatoire de Poisson)**.** Soit $\mathscr{L}$ un réseau de $\mathbb{R}^m$ et f une fonction raisonnable. On a :

$$f(\mathscr{L}):=\sum_{x\in\mathscr{L}} f(x)=\det\mathscr{L}^\vee\cdot\sum_{x\in\mathscr{L}^\vee}\hat{f}(x)=\det\mathscr{L}^\vee\cdot\hat{f}(\mathscr{L}^\vee).$$

6.3. **Distributions Gaussiennes.** Il existe plusieurs normalisation pour les fonctions gaussiennes. Dans ce cours, on fait un choix permettant de limiter l'apparition de facteurs $\sqrt{\pi}$ dans les quantités les plus utilisées, mais il n'est jamais possible de s'en débarasser.

Définition 6.7. La fonction Gaussienne centrée en $c\in\mathbb{R}^m$ et d'écart-type $s>0$ est définie pour tout $x\in\mathbb{R}^m$ par

$$\rho_{c,s}(x)=\exp\left(-\pi\frac{\|x-c\|^2}{s^2}\right).$$

Lorsque $c=0$ on écrira simplement ρ_s. On a $\int_{\mathbb{R}^m}\rho_{c,s}(x)dx=s^m$, ce qui est facile à montrer si on connait le résultat en dimension 1. Ceci permet de définir des variables aléatoires Gaussiennes continues $X\leftarrow\mathscr{N}_{c,s}$ en dimension m comme ayant pour densité $\mathscr{N}_{c,s}(x)=\rho_{c,s}(x)/s^m$. Si $X\leftarrow\mathscr{N}_{c,s}$, on a $\mathbb{E}[X]=c$ et $\mathrm{Cov}[X]=\mathrm{diag}(s^2)$.

Exercice 6.8. Démontrer les égalités annoncées (masse Gaussienne sur $\mathbb{R}^m$, espérance et matrice de covariance).

Comme annoncé, les Gaussiennes sont des fonctions propres de la transformée de Fourier.

Proposition 6.9. On a $\hat{\rho_s}=s^m\rho_{1/s}$.

Démonstration. L'astuce principale consiste à "compléter le carré" :

$$\begin{aligned}\hat{\rho_s}(\xi)=\int_{\mathbb{R}^m}\rho_s(x)e^{-2i\pi\langle x,\xi\rangle}dx&=\int_{\mathbb{R}^m}\exp\left(-\pi\left(\left\|\frac{x}{s}\right\|^2+2i\langle x,\xi\rangle\right)\right)dx\\&=\int_{\mathbb{R}^m}\exp\left(-\pi\left(\left\|\frac{x}{s}+is\xi\right\|^2+s^2\|\xi\|^2\right)\right)dx\\&=s^m\exp(-\pi s^2\|\xi\|^2)\cdot\int_{\mathbb{R}^m}\exp(-\pi\|x+is\xi\|^2)dx,\end{aligned}$$

où on a fait le changement de variable $x\to x/s$ à la dernière ligne. Il reste à justifier que la dernière intégrale vaut 1, ce qui se révèle plus suant que prévu à cause de la partie imaginaire (mais classique donc je le documente). Remarquons déjà que :

$$\int_{\mathbb{R}^m}\exp(-\pi\|x+is\xi\|^2)dx=\prod_{j=1}^m\int_{\mathbb{R}}\exp(-\pi(x_j+is\xi_j)^2)dx_j,$$

et il suffit donc de montrer que chaque facteur vaut 1. Une façon de le faire est d'utiliser une intégrale de contour : la fonction $z\mapsto\exp(-\pi z^2)$ est entière, et donc son intégrale sur un chemin fermé du plan complexe est nulle. Un chemin utile ici est γ_R : on parcourt $[-R;R]$ de gauche à droite, puis sans perte de généralité $[R;R+is\xi_j]$ vers le haut [16], puis $[R+is\xi_j;-R+is\xi_j]$ vers la gauche (c'est cette partie qui nous intéresse), et on redescend le long de $[-R+is\xi_j,-R]$. On a donc

$$\int_{\gamma_R}\exp(-\pi z^2)dz=\int_{-R}^R e^{-\pi x^2}dx+\int_0^{s\xi_j}e^{-\pi(R+it)^2}idt-\int_{-R}^R e^{-\pi(x+i\xi_j)^2}dx-\int_0^{s\xi}e^{-\pi(-R+it)^2}idt=0.$$

Quand $R\to+\infty$, le premier terme devient $\rho_{0,1}(\mathbb{R})=1$, on a donc terminé si on montre que les deux termes correspondants aux chemins verticaux tendent vers 0. En utilisant que $|e^z|=e^{\mathrm{Re}(z)}$, on a

$$\left|\int_0^{s\xi_j}e^{-\pi(R+it)^2}idt\right|\leqslant\int_0^{s\xi_j}e^{-\mathrm{Re}(\pi(R+it)^2)}dt=e^{-\pi R^2}\int_0^{s\xi_j}e^{\pi t^2}dt,$$

qui tend bien vers 0 avec R. L'autre terme se majore identiquement. □

Définition 6.10. Soit $\mathscr{L}$ un réseau de $\mathbb{R}^m$. La distribution Gaussienne *discrète* de support $\mathscr{L}$ et de paramètres $c\in\mathbb{R}^m, s>0$ est définie par la densité

$$D_{\mathscr{L},c,s}(x)=\frac{\rho_{c,s}(x)}{\rho_{c,s}(\mathscr{L})}=\frac{\rho_s(x-c)}{\rho_s(\mathscr{L}-c)},\forall x\in\mathscr{L}.$$

Il s'agit donc d'une fonction Gaussienne restreinte à un réseau, et normalisée pour être une loi de probabilité.

16. Sinon, on parcourt $[R,R-is\xi_j]$ vers le haut. Le but est que la paramétrisation du segment soit croissante pour se simplifier une majoration, plus tard.

Discrète à quel point ? Avant de continuer vers des Gaussiennes supportées dans des réseaux euclidiens, on peut essayer de se faire une idée de la situation. Intuitivement, si un réseau a un grand volume et qu'on regarde des Gaussiennes centrées ($c = 0$) de petits paramètres s, il est peu probable que $D_{\mathscr{L},c,s}$ renvoie autre chose que 0. Bien que ce soit une distribution à part entière, elle ne correspond pas vraiment à l'idée qu'on se fait d'une Gaussienne. Cela suggère que même si la distribution est centrée, le comportement attendu n'arrive que lorsque s est assez grand pour "gommer" la discrétude du réseau sous-jacent.

Lorsque la distribution n'est pas centrée en un point du réseau, d'autres phénomènes apparaissent [17] dès la dimension 1. Pour $c \in \mathbb{R}$, il est équivalent de regarder $D_{\mathbb{Z},c,s}$ ou $D_{\mathbb{Z},\{c\},s} + \lfloor c \rceil$, autrement dit c'est surtout la classe c mod $\mathbb{Z}$ qui importe. Si maintenant l'écart-type s est petit et que $\{c\} \neq 0$, on voit non seulement une dyssymétrie, mais aussi que $\rho_s(\mathbb{Z} + \{c\}) < \rho_s(\mathbb{Z})$ — un phénomène qui s'atténue si s augmente. Autrement dit, les cosets de $\mathbb{Z}$ dans $\mathbb{R}$ ont des masses Gaussiennes assez différentes de celle de $\mathbb{Z}$ quand s est trop petit. De trop grandes disparités peuvent être détectées plus facilement par des tests statistiques, et c'est rarement une bonne idée en cryptographie de dévier d'un comportement "uniforme".

Autre exemple : si un réseau de rang 2 a des minima très déséquilibrés, il est clair que prendre s proche de λ_1 va privilégier les points du réseau dans la direction d'un vecteur atteignant λ_1, tandis qu'on s'attend à ce qu'une Gaussienne s'étale de la même manière [18] dans tous l'espace. Ceci suggère que s devrait être du même ordre de grandeur que le dernier minimum du réseau pour que cette Gaussienne discrète ressemble un peu à une Gaussienne.

On va maintenant formaliser la situation et mettre en avant un nouvel invariant permettant de quantifier la "discrétude" d'un réseau vis-à-vis d'une Gaussienne : *le paramètre de lissage* [19] d'un réseau. Nous aurons besoin d'un peu de dualité, et d'un peu d'analyse harmonique.

6.4. **Lissage d'un réseau.** L'intuition est la suivante : ne pas pouvoir distinguer les cosets revient à dire que la distribution $\mathscr{N}_{c,s}$ mod $\mathscr{L}$ devrait être proche de la distribution uniforme sur le groupe compact $\mathbb{R}^m/\mathscr{L}$. Cette dernière donne (informellement) le même poids $(\det \mathscr{L})^{-1}$ à tous les cosets, et on espère donc que tous les $\rho_{s,c}(\mathscr{L})$ soient proches de cette valeur.

C'est ici que la Formule de Poisson (Théorème 6.6) va servir. D'abord, elle dit la masse Gaussienne est maximisée sur le réseau, ou autrement dit, que les cosets d'un réseau sont moins "lourds".

Proposition 6.11. Soit $\mathscr{L}$ un réseau et $c \in \mathrm{span}_{\mathbb{R}}(\mathscr{L})$. On a $\rho_s(c + \mathscr{L}) \leqslant \rho_s(\mathscr{L})$.

Démonstration. La Formule de Poisson 6.6 et le Lemme 6.5 donnent

$$\begin{aligned} \rho_{c,s}(\mathscr{L}) = \sum_{x \in \mathscr{L}} \rho_s \circ \tau_c(x) &= \frac{s^m}{\det \mathscr{L}} \cdot \sum_{\xi \in \mathscr{L}^\vee} \rho_{1/s}(\xi) e^{2i\pi\langle \xi, c\rangle} \\ &\leqslant \frac{s^m}{\det \mathscr{L}} \cdot \sum_{\xi \in \mathscr{L}^\vee} \rho_{1/s}(\xi) \\ &= \rho_s(\mathscr{L}), \end{aligned} \tag{1}$$

17. Et on le verra un jour sur des dessins.

18. En effet, c'est une fonction *radiale* : elle ne dépend que de la norme de sa variable.

19. Dans la littérature anglophone, on parle du "smoothing parameter".

où on a utilisé que $\rho_{1/s}$ est positive et l'inégalité triangulaire à la deuxième ligne, puis de nouveau la formule de Poisson. □

Corollaire 6.12. Soit $\mathscr{L}$ un réseau, $V = \mathrm{span}_{\mathbb{R}}(\mathscr{L})$ et P la projection orthogonale sur $V^{\perp}$. Pour $c \in \mathbb{R}^m$, on a $\rho_s(c+\mathscr{L}) \leqslant e^{-\pi\|P(c)\|^2/s^2}\rho_s(\mathscr{L})$.

Démonstration. Tout élément de $c+\mathscr{L}$ peut s'écrire $c+u = (c-P(c)+u)+P(c)$, où $u \in \mathscr{L}$, et par définition $x = c-P(c) \in V$. Avec le théorème de Pythagore, on a ensuite

$$\begin{aligned}\rho_s(c+\mathscr{L}) &= \sum_{u\in\mathscr{L}} \exp(-\pi\|P(c)\|^2/s^2)\exp(-\pi\|x+u\|^2/s^2)\\ &= \exp(-\pi\|P(c)\|^2/s^2)\cdot\rho_s(x+\mathscr{L}),\end{aligned}$$

et on conclut avec la Proposition 6.11. □

Ce corollaire montre que non seulement la masse des coset est plus petite, mais en plus qu'elle décroit exponentiellement vite à mesure qu'on s'éloigne de l'origine. On voit déjà ici que s doit être assez gros pour atténuer cette décroissance. Maintenant, si on réorganise la formule de Poisson, on obtient :

$$\rho_s(\mathscr{L}) = \frac{s^m}{\det\mathscr{L}}\cdot\rho_{1/s}(\mathscr{L}^{\vee}) \quad\Leftrightarrow\quad \mathscr{N}_s(\mathscr{L}) = \frac{\rho_{1/s}(\mathscr{L}^{\vee})}{\det\mathscr{L}}.$$

Notons que $\rho_{1/s}(\mathscr{L}^{\vee}) = 1+\sum_{x\in\mathscr{L}^{\vee}\setminus\{0\}}\exp(-\pi s^2\|x\|^2)$, et que la grande somme décroit très vite [20] quand s augmente. Comme on sait que les autres cosets auront une masse plus faibles, on peut chercher $s>0$ assez grand pour que $\rho_{1/s}(\mathscr{L}^{\vee}) = 1+\varepsilon$ pour un $\varepsilon>0$ moralement très petit.

Posons maintenant $\delta = \sum_{\xi\in\mathscr{L}^{\vee}\setminus\{0\}}\rho_{1/s}(\xi)e^{2i\pi\langle\xi,c\rangle}$, de sorte à ce que l'égalité (1) se réécrive $\mathscr{N}_{c,s}(\mathscr{L}) = \frac{1+\delta}{\det\mathscr{L}}$. On a $|\delta| \leqslant \rho_{1/s}(\mathscr{L}^{\vee})-1$, donc si on reprend le même s et le même ε, on a même $|\delta|\leqslant\varepsilon$. Il vient ce qu'on espérait, tous les cosets ont essentiellement la même masse :

$$\frac{1-\varepsilon}{\det\mathscr{L}} \leqslant \mathscr{N}_{c,s}(\mathscr{L}) \leqslant \mathscr{N}_s(\mathscr{L}) \leqslant \frac{1+\varepsilon}{\det\mathscr{L}}. \tag{2}$$

Définition 6.13. Soit $\mathscr{L}$ un réseau et $0<\varepsilon<1$. Le paramètre de ε-lissage de $\mathscr{L}$ est

$$\eta_{\varepsilon}(\mathscr{L}) = \inf\{s>0 \ :\ \rho_{1/s}(\mathscr{L}^{\vee}) \leqslant 1+\varepsilon\}.$$

Proposition 6.14. Soit $\mathscr{L}$ un réseau et $0<\varepsilon<1$. Si $s \geqslant \eta_{\varepsilon}(\mathscr{L})$, alors pour tout $c\in\mathbb{R}^m$, on a

$$\rho_s(\mathscr{L}+c) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right]\cdot\rho_s(\mathscr{L}).$$

Démonstration. Le preuve a conduit à l'encadrement (2), qu'il suffit ensuite de réorganiser. □

On peut aussi montrer qu'au dessus du lissage, une Gaussienne discrète $D_{\mathscr{L},c,s}$ a un comportement probabiliste très proche d'une Gaussienne continue : son espérance est "presque" c, sa variance est "presque" s^2 et elle se concentre "presque" aussi bien que la Gaussienne (voir par exemple le très beau papier de Micciancio et Regev [MR07]).

20. Cette situation est une bonne illustration que lorsqu'une fonction est "bien étalée", sa transformée de Fourier a tendance à se concentrer en 0, ce qui est particulièrement visible sur les Gaussiennes.

Remarque 6.15. Le point de départ était que, si tous les cosets ont à peu près la même masse, la distribution $\mathscr{N}_s$ mod $\mathscr{L}$ et l'uniforme $\mathscr{U}$ sur le groupe compact $G = \mathbb{R}^m/\mathscr{L}$ sont très proches l'une de l'autre. Ceci peut se formaliser. On rappelle que le poussé en avant d'une mesure sur un groupe quotient est la mesure du coset correspondant au représentant, et que si G est compact, son volume est fini donc on peut toujours renormaliser une mesure sur G en une loi de probabilité.

Définition 6.16. La distance statistique entre 2 fonctions $f, g : G \to \mathbb{R}$ est définie par $\Delta(f,g) = \frac{1}{2}\int_G |f(x) - g(x)| d\mu(x)$, où $d\mu$ est la mesure de probabilité sur G.

Proposition 6.17. Si $s \geqslant \eta_\varepsilon(\mathscr{L})$ alors $\Delta(\mathscr{N}_{c,s} \bmod \mathscr{L}, \mathscr{U}) \leqslant \varepsilon/2$.

Démonstration. On prend $f = \mathscr{N}_{c,s} \bmod \mathscr{L}$ et g l'uniforme sur G. L'hypothèse sur s nous permet d'utiliser l'encadrement (2), ce qui donne immédiatement le résultat. □

Remarque 6.18. Si on a une inclusion $\mathscr{L}' \subset \mathscr{L}$ de réseaux de même rang, on peut de la même manière montrer qu'au dessus du lissage de $\mathscr{L}'$, une Gaussienne prise sur $\mathscr{L}$ et réduite $\bmod \mathscr{L}'$ a l'air uniforme dans le groupe fini $\mathscr{L}/\mathscr{L}'$.

On termine cette section avec quelques propriétés du paramètre de lissage. Le résultat suivant traduit son *homogénéité*.

Lemme 6.19. Pour tout réseau $\mathscr{L}$, $a \in \mathbb{R}^*$ et $\varepsilon > 0$, on a $\eta_\varepsilon(a\mathscr{L}) = |a|\eta_\varepsilon(\mathscr{L})$. Si $u \in \mathbb{R}^m \setminus \{0\}$, on a $\eta_\varepsilon(\mathbb{Z}u) = \|u\|\eta_\varepsilon(\mathbb{Z})$.

Démonstration. Il s'agit d'appliquer le Lemme 6.3 pour avoir les duaux. Par exemple, pour la première propriété, on a $\rho_{1/s}((a\mathscr{L})^\vee) = \rho_{1/s}(\frac{1}{a}\mathscr{L}^\vee) = \rho_{|a|/s}(\mathscr{L}^\vee)$, cette dernière somme valant $1+\varepsilon$ si et seulement si $s/|a| = \eta_\varepsilon(\mathscr{L})$. La deuxième propriété se montre similairement. □

Il existe plusieurs types d'estimation pour la paramètre de lissage. Les estimations *abstraites* (ou théoriques) font généralement intervenir des quantités fondamentales difficiles à estimer elles-mêmes, comme les minima d'un réseaux ou son rayon de recouvrement. Elles sont généralement les meilleures quand on peut leur donner une valeur. Par exemple, on sait que $\eta_\varepsilon(\mathscr{L}) = \lambda_n(\mathscr{L}) \cdot O\big(\sqrt{\log(n/\varepsilon)}\big)$.

Proposition 6.20 (Micciancio-Regev)**.** Pour tout $\varepsilon > 0$, on a $\eta_\varepsilon(\mathbb{Z}^n) \leqslant \sqrt{\log(2n(1+1/\varepsilon))/\pi}$. Plus généralement, pour tout réseau $\mathscr{L}$ de rang n, on a $\eta_\varepsilon(\mathscr{L}) \leqslant \lambda_n(\mathscr{L}) \cdot \sqrt{\log(2n(1+1/\varepsilon))/\pi}$.

Les estimations *concrètes* font intervenir les bases d'un réseau, et des quantités liées à leur représentation matricielle. Elles sont d'une part explicite si la base est donnée, bien qu'estimer des quantités fondamentales de matrices aléatoires ne soit généralement pas un problème simple [21]. D'autre part, elles sont généralement *effectives* : il existe un algorithme d'échantillonnage Gaussien qui atteint la borne [22].

Proposition 6.21 (Gentry-Peikert-Vaikunthanatan)**.** Pour tout $0 < \varepsilon < \frac{1}{n}$, pour tout réseau $\mathscr{L}$ de rang n, on a $\eta_{2\varepsilon}(\mathscr{L}) \leqslant \min_{\substack{(b_1,\dots,b_n) \\ \text{base de } \mathscr{L}}} \max_{1\leqslant i\leqslant n} \|b_i^*\| \cdot \eta_{\varepsilon/n}(\mathbb{Z})$.

21. Loin s'en faut.

22. En particulier, un problème intéressant est d'arriver à échantillonner efficacement très proche du lissage d'un réseau

Preuve différente de l'originale. On propose une preuve utilisant d'une part l'identification des projections orthogonales comme réseaux quotients, d'autre part l'argument que les cosets ont tous une masse Gaussienne plus petite que celle de leur réseau. Soit $b_1,\dots,b_n$ une base de $\mathscr{L}$. Posons $\mathscr{L}' = \mathscr{L}(b_1,\dots,b_{n-1})$, P la projection orthogonale sur $\mathrm{span}_{\mathbb{R}}(\mathscr{L}')^{\perp}$, et soit $x \in \mathscr{L}\setminus\mathscr{L}'$. Par construction, l'ensemble $P(\mathscr{L}')$ est le réseau $\mathbb{Z}b_n^*$. D'autre part, on a une correspondance bijective entre chaque coset $x+\mathscr{L}'$ et $P(x)$: autrement dit, on peut identifier $\mathscr{L}/\mathscr{L}'$ avec $\mathbb{Z}b_n^*$. Par le Corollaire 6.12, on a $\rho_s(x+\mathscr{L}') \leqslant e^{-\pi\|P(x)\|^2}\cdot\rho_s(\mathscr{L}')$, et en sommant sur tout $\mathbb{Z}b_n^*$, on obtient

$$\rho_s(\mathscr{L}) \leqslant \rho_s(\mathbb{Z}b_n^*)\cdot\rho_s(\mathscr{L}').$$

On peut recommencer le même procédé avec $\mathscr{L}'$ et ainsi de suite : par induction, on obtient $\rho_s(\mathscr{L}) \leqslant \prod_{i=1}^n \rho_s(\mathbb{Z}b_i^*)$. En appliquant la formule de Poisson et le Lemme 6.3, cette inégalité est équivalente à

$$\rho_{1/s}(\mathscr{L}^\vee) \leqslant \prod_{i=1}^n \rho_{\|b_i^*\|/s}(\mathbb{Z}).$$

En prenant $s \geqslant \max_i \|b_i^*\|\cdot\eta_{\varepsilon/n}(\mathbb{Z})$, il vient $\rho_{1/s}(\mathscr{L}) \leqslant (1+\varepsilon/n)^n \leqslant (1-\varepsilon)^{-1} \leqslant 1+2\varepsilon$. On conclut en notant que le choix de la base était arbitraire. □

6.5. **L'algorithme de Klein.** On a vu que l'algorithme *Nearest Plane* de Babai permet de résoudre *déterministiquement* des problèmes $\mathbf{CVP}_\gamma$ par décodage, le facteur d'approximation dépendant de la qualité de la base donnée en entrée. Un point de vue utile pour la suite est de considérer qu'un échantillonneur Gaussien dans un réseau est un algorithme *randomisé* pour ce problème : plus précisément, si on sait randomiser correctement le décodage avec un bruit Gaussien, on peut construire un échantillonneur.

On va rapidement voir que le problème se ramène à la possibilité d'échantillonner des *entiers* Gaussiens, c'est-à-dire, qu'il suffit d'avoir un tel algorithme en dimension 1, mais aussi qu'on ne va avoir qu'une distribution statistiquement très proche de celle qu'on veut. On verra des approches pour obtenir des entiers Gaussiens à la fin du chapitre ; pour l'instant, on considèrera qu'on a accès à une boite noire $\mathsf{B}(\mathbb{Z},c,s)$ qui nous renvoie un entier $z \leftarrow D_{\mathbb{Z},c,s}$ chaque fois qu'on lui demande.

Algorithm 5: Echantillonneur de Klein

Entrées: une base $(b_i)_{i\leqslant m}$ d'un réseau et sa Gram-Schmidt $(b_i^*)_{i\leqslant m}$;

un paramètre $s>0$ et un vecteur $t\in\mathbb{R}^m$.

Sortie: un vecteur $v\in\mathscr{L}(b_1,\dots,b_m)$.

Calculer $s_i := \dfrac{s}{\|b_i^*\|^2}$;

Poser $v=0$;

pour $i = n$ *jusqu'à* 1 **faire**

  $c_i \leftarrow \dfrac{\langle t, b_i^*\rangle}{\|b_i^*\|^2}$;

  $z_i \leftarrow \mathsf{B}(\mathbb{Z},c,s_i)$ //$z \leftarrow D_{\mathbb{Z},c,s_i}$;

  $v_i \leftarrow v+zb_i$ et $t_i \leftarrow t-zb_i$;

fin pour

Renvoyer v;

L'intuition pour la première étape est que, si on échantillonnait des Gaussiennes toutes de la même taille, la distribution en sortie serait fortement biaisée dans les directions des b_i^* et donc pas vraiment radiale. On veut donc renormaliser par les $\|b_i^*\|$, mais cela fait diminuer l'écart-type qui pourrait être trop petit pour lisser les réseaux $\mathbb{Z}b_i^*$ sur chaque axe. Ceci explique aussi la condition de lissage : on doit avoir assez de "place" pour renormaliser sans que le réseau biaise la distribution de sortie.

Proposition 6.22. Si $\varepsilon \leqslant 1/2\sqrt{n}$ et $s \geqslant \max_i \|b_i^*\| \cdot \eta_\varepsilon(\mathbb{Z})$, l'algorithme 5 renvoie un vecteur dont la distribution $\mathscr{D}$ est à distance statistique $n\varepsilon$ de $D_{\mathscr{L},t,s}$.

Démonstration. Il est clair que l'algorithme renvoie des vecteurs de $\mathscr{L}$. Pour comprendre la preuve, on va la dérouler en dimension 2. Notons $t = \mathbf{B}^*(t_1,t_2)$ pour avoir $c_1 = t_1 - z_2\mu_{2,1}$ et $c_2 = t_2$, où $\mu_{2,1} = \langle b_2, b_1\rangle / \|b_1\|^2$ est la coordonnée Gram-Schmidt de b_2 sur b_1. On constate alors que l'algorithme est construit pour satisfaire l'égalité

$$v - t = \mathbf{B}(z_1,z_2) - \mathbf{B}^*(t_1,t_2) = \mathbf{B}^*(\mathbf{U}(z_1,z_2) - (t_1,t_2)) = \mathbf{B}^*(z_1 - c_1, z_2 - c_2). \tag{3}$$

En dimension arbitraire, cette idée restera la même. D'autre part, la probabilité d'avoir v en sortie est

$$P(z_1,z_2) = D_{\mathbb{Z},s_2,c_2}(z_2) \cdot D_{\mathbb{Z},s_1,c_1}(z_1) = \frac{\rho_{s_2}(z_2 - c_2) \cdot \rho_{s_1}(z_1 - c_1)}{\rho_{s_2}(\mathbb{Z} - c_2) \cdot \rho_{s_1}(\mathbb{Z} - c_1)}. \tag{4}$$

Avec les propriétés de l'exponentielle, le théorème de Pythagore, puis l'identité (3), le numérateur N se réécrit alors

$$N = \exp\left(-\frac{\pi}{s^2} \cdot \|\mathbf{B}^*(z_1 - c_1, z_2 - c_2)\|^2\right) = \rho_s(v - t).$$

Par contre, on n'obtient pas grand chose de concluant en travaillant avec le dénominateur, et il va falloir un argument supplémentaire. Commençons par réécrire l'égalité (4) comme

$$P(z_1,z_2) = D_{\mathscr{L},t,s}(v) \cdot \frac{\rho_s(\mathscr{L} - t)}{\rho_{s_1}(\mathbb{Z} - t_1 + z_2\mu_{2,1})\rho_{s_2}(\mathbb{Z} - t_2)}, \tag{5}$$

pour observer qu'on a gagné si on peut montrer que la fraction de droite est proche de 1 quel que soit z_2. C'est ici qu'on utilise un argument de lissage : par hypothèse et par le Lemme 6.19, on a $s \geqslant \|b_1\| \cdot \eta_\varepsilon(\mathbb{Z}) = \eta_\varepsilon(\mathbb{Z}b_1)$. On peut donc appliquer la Proposition 6.14 pour obtenir

$$P(z_1,z_2) \in \left[1, \frac{1+\varepsilon}{1-\varepsilon}\right] \cdot D_{\mathscr{L},t,s}(v) \cdot \frac{\rho_s(\mathscr{L} - t)}{\rho_{s_1}(\mathbb{Z})\rho_{s_2}(\mathbb{Z} - t_2)}.$$

L'intérêt est que la fraction de droite, α, ne dépend plus de z_2. En sommant sur toutes les paires (z_1,z_2), on obtient $\frac{1-\varepsilon}{1+\varepsilon} \leqslant \alpha \leqslant 1$, puis finalement l'encadrement

$$P(z_1,z_2) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] \cdot D_{\mathscr{L},t,s}(v).$$

Ceci implique $|P(z_1,z_2) - D_{\mathscr{L},t,s}(v)| \leqslant \frac{2\varepsilon}{1-\varepsilon} D_{\mathscr{L},t,s}(v)$, et en sommant encore une fois sur toutes les possibilités, on obtient $\Delta(\mathscr{D}, D_{\mathscr{L},t,s}) \leqslant \frac{\varepsilon}{1-\varepsilon}$. La preuve pour n arbitraire est essentiellement identique (exercice !). □

Cas général, ira en Annexe. En notant $t = \mathbf{B}^*(t_1,\ldots,t_n)$ et μ_{ji} les entrées de $\mathbf{U}$, on a $c_n = t_n$ et $c_{n-i} = t_{n-i} - \sum_{j=0}^{i-1} z_j\mu_{n-j,n-i}$. Ceci donne $v - t = \mathbf{B}^*(z_i - c_i)$, et donc $\rho_s(v-t) = \prod_{i=1}^n \rho_{s_i}(z_i - c_i)$. Le probabilité d'obtenir v en sortie est celle d'obtenir $z_1,\ldots,z_n$:

$$P(z_1,\ldots,z_n) = \frac{\rho_s(v-t)}{\prod_{i=1}^n \rho_{s_i}(\mathbb{Z} - c_i)} = \alpha \cdot D_{\mathscr{L},t,s}(v),$$

où $\alpha = \frac{\rho_s(\mathscr{L}-t)}{\prod_{i=1}^n \rho_s(\mathbb{Z}b_i^* - c_i b_i^*)}$. Pour $1 \leqslant i \leqslant n-1$, les cosets au dénominateur dépendent de z_i. Comme on a $s \geqslant \eta_\varepsilon(\mathbb{Z}b_i^*)$ par hypothèse, la Proposition 6.14 donne

$$P(z_1,\dots,z_n) \in \left[1, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{n-1}\right] \cdot \alpha \cdot D_{\mathscr{L},t,s}(v).$$

Avec l'argument de sommation précédent, on a $\alpha \in [(\frac{1-\varepsilon}{1+\varepsilon})^{n-1}, 1]$ et on obtient l'encadrement

$$P(z_1,\dots,z_n) \in \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{n-1}, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{n-1}\right] \cdot D_{\mathscr{L},t,s}(v).$$

A ce stade, on est déjà convaincu que si ε est assez petit, l'algorithme echantillonne essentiellement la distribution ciblée. En écrivant $(\frac{1+\varepsilon}{1-\varepsilon})^{n-1} = \exp((n-1)(\log(1+\varepsilon) - \log(1-\varepsilon))$, et en utilisant les développement de Taylor classiques, l'encadrement ci-dessus et le choix pour ε implique que $|P(z_1,\dots,z_n) - D_{\mathscr{L},t,s}(v)| \leqslant 2n\varepsilon \cdot D_{\mathscr{L},t,s}(v)$. Un dernier argument de sommation conclut pour la distance statistique. □

7. Exemple en pratique d'une signature fondées sur les réseaux Euclidiens

7.1. **Contexte : la cryptographie post-quantique.** Les cryptosystèmes à clé publique actuellement en utilisation dans **tous** les protocoles de communications reposent sur la difficulté de factoriser des grands entiers ou de calculer des logarithmes discrets dans des grands groupes commutatifs. Par *difficile* ici, il faut comprendre qu'en l'état de l'art, les meilleurs algorithmes connus pour résoudre ces problèmes sont de complexité exponentielle en la taille (en bit) des données manipulées. Concrètement, il est inutile de les exécuter même sur les meilleurs supercalculateurs actuels pour essayer de casser un cryptosystème en activité : notre système solaire aura depuis longtemps disparu quand ils renverront [23] les clés secrètes visées.

En 1984, Shor décrit un algorithme *quantique*, c'est-à-dire devant s'exécuter sur un ordinateur quantique et utilisant des techniques de calcul quantique, qui résout la factorisation et les calculs de logarithmes discrets en temps polynomial. En termes concrets, cela signifie qu'à capacités de calcul *quantique* équivalentes à nos capacités *classiques* actuelles, on casserait rapidement [24] ces mêmes cryptosystèmes, dans des ordres de grandeurs de temps possiblement très dangereux pour la sécurité des communications.

Les progrès en calcul quantique ont été réguliers : on est passé d'un objet à l'intérêt purement théorique dans les années 90 à des résultats expérimentaux probants sur de véritables machines quantiques ces dernières années. La question soulevée aux alentours de 2015 dans les communauté cryptographique et de la cyber-sécurité a alors été : combien de temps avant de voir émerger une machine quantique assez puissante ? Il n'y a pour l'instant pas [25] de réponse ni de consensus. Quoi qu'il en soit, cela créerait un certain chaos si cela arrivait plus tôt que "prévu" — typiquement, avant la date de péremption de données actuellement chiffrées à très haut niveau de confidentialité, comme des codes d'engins nucléaires. Plutôt que d'attendre sans rien faire,la communauté de la cryptographie a choisi

23. Du coup, ils seront interrompus en fait, on n'aura jamais ces clés.

24. La réelle complexité du calcul est délicate à estimer, ce qui est entre dû à l'absence de référentiel technologique.

25. "Des experts" attendent une telle machine sous une dizaine d'année ; de manière plus mesurée pour "d'autres experts", il pourrait être raisonnable d'envisager la situation sous 20, 30 ans. L'emploi des guillemets se justifie par la difficulté (l'impossibilité ?) à savoir lesquels, d'experts. Certains physiciens identifiés estiment que l'ordinateur quantique à cette échelle de puissance restera à jamais du domaine du fantastique. Dans tous les cas, on peut s'interroger sur les intérêts (financiers, stratégiques, sécuritaires) derrière ce type de postulat.

de répondre à cette question en concevant de nouveaux cryptosystèmes à clé publique (conjecturés) capable de résister à de potentiels attaquants quantiques, et donc qualifiés de *post-quantique*.

7.2. **La compétition du NIST.** Une illustration de cette décision est l'appel à standardisation organisé entre tois tours de 2017 à 2022 par le *National Institute of Standards and Technologies* (NIST) : plus de soixantes propositions réparties entre mécanismes d'échange de clés (KEM) et signatures numériques ont été soumises à cette compétition, déclenchant en parallèle un large effort d'analyse théorique, de cryptanalyse et de mise en pratique de ces schémas. De nouvelles hypothèses de difficulté ont été mis en avant, impliquant :

— les réseaux euclidiens (ce qui nous intéresse) ;

— les codes correcteurs ;

— les isogénies entre courbes elliptiques ;

— les systèmes multivariés ;

— d'autres choses plus exotiques.

Ayant vocation à être utilisés en pratique, les schémas proposés se doivent d'être *concrètement efficaces* [26]. Ici, il ne s'agit plus d'être simplement "polynomial", et d'ailleurs tous vainqueurs de la compétition ont une complexité quasi-linéraire en temps et en mémoire. Le sens d'efficacité ici devient beaucoup plus concret, et dépend du contexte d'utilisation, mais cette introduction restera assez vague. On veut que les temps d'exécution des algorithmes soient comparable à ceux des cryptosystèmes qu'on utilise déjà. L'intervention de ces deeniers étant absolument transparente pour un utilisateur, cela ne doit pas changer en passant au post-quantique. D'autre part, leur consommation de bande passante doit aussi être minimale [27], quelques kilobytes au grand maximum pour passer les clés et les signatures doivent suffire.

A l'issue de la compétition, trois schémas ont été retenus [28] pour standardisation dans les prochaines années. Il se trouve qu'ils sont tous fondés sur des problèmes difficiles de réseaux euclidiens :

— l'échange de clé KYBER et la signature DILLITHIUM, voir ce site ;

— la signature FALCON.

Fondamentalement, ce sont des instantiations des primitives théoriques de réseaux euclidiens aperçues dans le cours. Le premier est un chiffrement "à la Regev", le second une signature de type "Fiat-Shamir avec répétitions" et le dernier une version de "hasher-signer" sur les réseaux euclidiens. Leur point commun est de faire intervenir des *réseaux algébriques*, correspondant à des modules libres de petit rang sur des anneaux d'entiers cyclotomiques. L'objectif n'est pas ici de faire un cours de théorie algébrique des nombres, on se restreindra donc au minimum syndical pour comprendre les principes (plus tard, il y aura une annexe plus fournie sur le sujet). D'autre part, on se concentrera sur FALCON dans ces notes.

26. On veut aussi que leurs implémentations soit résilientes face aux attaques par canaux auxiliaires, mais ceci nous entrainerait bien loin de ces notes de cours.

27. Imaginer le stress supplémentaire du réseau global si, du jour au lendemain, certains sites de vente en ligne (par exemple) devaient échanger avec *chaque* utilisateur des données d'identifications 10 fois plus grandes lors d'un évènement de grande ampleur. Et on ne parle pas encore d'écologie.

28. D'autres schémas sont considérés comme solutions conservatrices ou alternatives, et seront standardisés dans un second temps.

7.3. **Signatures “à la GPV” en bref.** On rappelle brièvement les principes des signatures hasher-signer pour les réseaux euclidiens :

— un message M est hashé en un vecteur $m = \mathtt{H}(M)$ dans l'espace ambiant d'un réseau public $\mathscr{L}$;

— on calcule un point $v \in \mathscr{L}$ *assez proche* de $\mathtt{H}(M)$ et la signature est $s = \mathtt{H}(M) - v$.

— une paire (M,s) est valide si et seulement si $\mathtt{H}(M) - s \in \mathscr{L}$ **et** $\|s\|$ est assez petite.

Signer est clairement une instance d'un problème **CVP**, qui doit donc être dure à résoudre efficacement pour un attaquant. D'après la Section 5, il est possible de le faire avec une base $\mathbf{B}$ de vecteurs courts de $\mathscr{L}$, une telle base devrait donc rester secrète, connue seulement du signataire et être difficile à calculer sans connaissances supplémentaires. Inversement, n'importe qui doit pouvoir facilement vérifier l'appartenance de $\mathtt{H}(M) - s$ au réseau, et on doit pour cela publier une (mauvaise) base $\mathbf{A}$. Pour donner vie à ce type de schéma, il faut donc :

(1) une procédure efficace pour générer des paires $(\mathbf{A},\mathbf{B})$ de bases de réseaux, où $\mathbf{B}$ est appelée une *trappe* et est composée de vecteurs courts ;

(2) une procédure efficace exploitant $\mathbf{B}$ pour signer, c'est-à-dire, résoudre des instances aléatoires de **CVP**.

Le paradigme de Gentry, Peikert et Vaikuntanathan (GPV) [GPV08] consiste à instancier les trois algorithmes `KeyGen`, `Sign` et `Verif` d'un schéma de signature à l'aide de réseaux q-aire $\Lambda_q^\perp(\mathbf{A})$, où q est un entier premier. Ici, $\mathbf{A}$ n'est pas tout à fait une base de $\mathscr{L} = \Lambda_q(\mathbf{A})^\perp$, mais si $\mathbf{B}$ en est une, on a $\mathbf{AB} = 0 \bmod q$, ce qui rend facile la vérification [29] de l'appartenance au réseau. Nous allons dans un premier temps supposer que `KeyGen` renvoie une paire $(\mathbf{A},\mathbf{B})$ comme décrite ci-dessus, sans détailler comment. Le reste du schéma prend la forme suivante :

(1) $\mathtt{Sign}(\mathbf{B},M)$ utilise une fonction de hash cryptographique pour calculer $m = \mathtt{H}(M)$ dans l'espace engendré par $\mathscr{L}$, puis en calcule une préimage c, c'est-à-dire qu'on doit avoir $\mathbf{A}c = m$. Une sous-procédure $\mathtt{Sample}(\mathbf{B},c)$ trouve un vecteur aléatoire $v \in \mathscr{L}$ proche de c grâce à $\mathbf{B}$. La signature renvoyée est $s = c - v$.

(2) $\mathtt{Verif}(\mathbf{A},M,s,\beta)$ calcule $m = \mathtt{H}(M)$ et $c' = \mathbf{A}s \bmod q$. Si s est une signature valide, alors elle s'écrit $s = c - v$ avec $v \in \mathscr{L}$ donc $\mathbf{A}s = \mathbf{A}c \bmod q$. On vérifie donc d'abord que $c' = m$, puis ensuite que $\|s\| \leqslant \beta$, où β est une borne publique assez petite pour qu'il soit difficile de signer sans connaître $\mathbf{B}$.

Comme vu dans le cours, si le problème ISIS est difficile, ce schéma de signature est sEU-CMA (strong-Existential Unforgeability against Chosen Message Attack). Cependant, il s'agit d'une preuve de sécurité *théorique*, et les paramètres qu'elle implique (dimension des réseaux, taille de q particulièrement) sont très loin de permettre l'efficacité attendue. Pour s'en convaincre, la matrice $\mathbf{A}$ par exemple devrait être rectangulaire $n \times m$, sans technique supplémentaire il faudrait donc $nm \log q$ bits pour la stocker, et $O(nm)$ opérations pour les multiplications. En pratique $n < m$ seraient de l'ordre du millier (et plus !), et il n'y a quasiment pas d'espoir d'amélioration substantielle si on en reste là. Pour obtenir des instanciations efficaces, KYBER, DILLITHIUM et FALCON se restreignent à des réseaux algébriques, possédant des symétries naturelles supplémentaires qui permettent à la fois d'améliorer la vitesse des opérations et de réduire la taille des données. En particulier, FALCON fait intervenir des réseaux NTRU [DLP14] (vus en TD, et qui seront présentés à nouveau plus bas).

29. Pour cette raison, de telles matrices sont aussi appelées matrices de vérification en théorie des codes.

En pratique, la deuxième difficulté à surmonter est l'instanciation de la procédure `Sample`. L'algorithme "Round-off" (Algorithme 3) était utilisé à l'origine [HHP+03], mais son déterminisme est une faiblesse du schéma. En effet, chaque signature donne de l'information sur la trappe utilisée pour la générer — plus précisément, sur le domaine fondamental associé — ce qui permet de la reconstruire avec des techniques d'apprentissage statistique [NR06, DN12]. La procédure `Sample` ne doit donc pas fuiter d'information sur la trappe. Depuis [GPV08], on l'instancie à l'aide de (variante de) l'algorithme de Klein (Algorithme 5). Les paramètres de la distribution Gaussienne échantillonnée (support, centre, écart-type) sont tous publics, et un attaquant observant des signatures n'apprend rien sur la trappe utilisée pour les générer.

7.4. **Réseaux NTRU en bref.** *Je fais le choix de ne pas aborder les plongements complexes des corps de nombres pour pouvoir me concentrer sur les idées derrière la conception de* FALCON *dans cette partie. Les notions associées de théorie algébrique des nombres peuvent aider à assainir la fin de cette section, mais cela rajouterait du matériel peu utile pour comprendre ce qu'il se passe en plus d'être équivalent moralement au point de vue "matrice de multiplication". Ce sera proposé dans une annexe future.*

Pour le reste de ce chapitre, on fixe $d = 2^\ell$ et ζ une racine primitive $2d$-ème de l'unité, de polynôme minimal $\Phi_d = X^d + 1$. On note $K = \mathbb{Q}[\zeta] \simeq \mathbb{Q}[X]/(X^d+1)$ le corps cyclotomique correspondant, et $R = \mathbb{Z}[\zeta]$ son anneau d'entiers algébriques. Tout élément de K ou R s'écrit donc $f = \sum_i f_{i<d}\zeta^i$, avec les f_i rationnels ou entiers, traduisant que K est un $\mathbb{Q}$-espace vectoriel de dimension d (et que R est un $\mathbb{Z}$-module libre de rang d). Ceci permet d'identifier f avec le vecteur $\mathbf{f} = (f_0, \dots, f_{d-1})$ de $\mathbb{R}^d$, et de parler de sa norme euclidienne $\|f\|^2 = \sum_i f_i^2$. On l'étend à des vecteurs de K^2 par $\|(f,f')\|^2 = \|f\|^2 + \|f'\|^2$. L'application $x \mapsto ax$ de multiplication par $a \in K$ est $\mathbb{Q}$-linéaire, et peut donc se représenter par une matrice dans une base de K sur $\mathbb{Q}$. Au vu de la définition de Φ_d, dans la base $1, \zeta, \dots, \zeta^{d-1}$, il s'agit d'une matrice *néga-cyclique* et pour $a = \sum_i a_i\zeta^i$, on la notera

$$[a] = \begin{bmatrix} a_0 & -a_{d-1} & \dots & -a_1 \\ a_1 & a_0 & \dots & -a_2 \\ \vdots & & \ddots & \vdots \\ a_{d-1} & a_{d-2} & \dots & a_0 \end{bmatrix}.$$

On vérifie que $[a+b] = [a] + [b]$ et $[ab] = [a][b]$ pour tout $a, b \in K$.

On fixe un entier q premier, et soient $f, g \in R$ tel que f soit inversible modulo q, de sorte que $h = g/f \bmod q$ est bien défini. Le module NTRU associé à h est alors l'ensemble

$$\Lambda_{\text{NTRU}}(h) = \{(u,v) \in R^2 \ : \ uh - v \equiv 0 \bmod q\}.$$

Il correspond à un réseau euclidien q-aire associé à la matrice $[[h], -\mathrm{Id}_d]$, qui dépend uniquement de h. On l'appelle réseau NTRU et on le notera

$$\mathscr{L}_{\text{NTRU}}(h) := \Lambda_q^\perp([[h], -\mathrm{Id}_d]) = \{(u,v) \in \mathbb{Z}^{2d} \ : \ [h]u - v \equiv 0 \bmod q\}.$$

On rassemble maintenant les propriétés fondamentales de ces réseaux.

Proposition 7.1. Le réseau NTRU est de rang $2d$, de volume q^d, et admet des bases (sous forme matricielle)

$$\mathbf{B}_h = \begin{bmatrix} \mathrm{Id}_d & 0 \\ [h] & q\,\mathrm{Id}_d \end{bmatrix} \quad \text{et} \quad \mathbf{B}_{f,g} = \begin{bmatrix} [f] & [F] \\ [g] & [G] \end{bmatrix},$$

où $(F,G) \in R^2$ est tel que $fG - gF = q$.

Démonstration. L'application $\psi : \mathbb{Z}^{2d} \to (\mathbb{Z}/q\mathbb{Z})^d$ par $\psi(u,v) = [h]u - v \bmod q$ définit un homomorphisme surjectif de groupes abéliens. On a $\ker \psi = \mathscr{L}_{\mathrm{NTRU}}(h)$ par construction, donc $\mathscr{L}_{\mathrm{NTRU}}(h)$ est un sous-réseau de $\mathbb{Z}^{2d}$. D'autre part, il est de volume q^d (pourquoi ?). Il est clair que $\mathscr{L}_{\mathrm{NTRU}}$ contient le réseau $\mathscr{L}(\mathbf{B}_h)$ qui est lui aussi de volume q^d et de rang $2d$, et le Lemme 1.11 donne $\mathscr{L}_{\mathrm{NTRU}}(h) = \mathscr{L}(\mathbf{B}_h)$. Par définition de h, on a $(f,g) \in \Lambda_{\mathrm{NTRU}}(h)$, et comme c'est un R-module, les premières d colonnes de $\mathbf{B}_{f,g}$ sont dans $\mathscr{L}_{\mathrm{NTRU}}(h)$. On a aussi $f(Fh - G) \equiv Fg - fG \equiv 0 \bmod q$ par hypothèse, et comme f est inversible modulo q, ceci implique que $(F,G) \in \Lambda_{\mathrm{NTRU}}(h)$. Ainsi, les d dernières colonnes de $\mathbf{B}_{f,g}$ sont dans le réseau NTRU, et comme $\det \mathbf{B}_{f,g} = q^d$ (pourquoi ?), la matrice $\mathbf{B}_{f,g}$ décrit bien une base de $\mathscr{L}_{\mathrm{NTRU}}(h)$. □

D'après cette proposition et le théorème de Minkowski (Théorème 1.15), on a $\lambda_1(\mathscr{L}_{\mathrm{NTRU}}) \leqslant \sqrt{2dq}$. On s'attend en fait à ce que tous les minima d'un réseau NTRU "générique" soit en $O(\sqrt{q})$, traduisant qu'il devrait être bien équilibré. C'est cette quantité qui va guider les choix de paramètres à venir : quand on parlera d'un "petit" vecteur lié à $\mathscr{L}_{\mathrm{NTRU}}$, c'est que sa taille est relativement proche de $\sqrt{q}$.

7.5. **FALCON simplifié, en bref : "GPV efficace" sur les réseaux NTRU.** On décrit dans un premier temps comment le schéma de signature s'instancie pour les réseaux NTRU. Les paragraphes suivants en expliciteront les étapes. Pour le reste du document, quand on utilisera la notation `Sample`, ce sera pour désigner l'échantillonneur de Klein (Algorithme 5). On notera aussi $R_q = \{x = \sum_i x_i \zeta^i \in R \ : \ \max_i |x_i| \leqslant q/2\}$.

(1) $\mathtt{KeyGen}(q,d)$ renvoie :

- $h \in R_q$ correspondant à la matrice $\mathbf{A} = [[h], -\mathrm{Id}_d]$;
- la trappe $\mathbf{B}_{f,g}$ composée de $(f,g),(F,G) \in R_q^2$ qui doivent être petits tous les deux ;
- (une représentation secrète de) la Gram-Schmidt $\widetilde{\mathbf{B}}_{f,g} = [\widetilde{b}_1, \ldots, \widetilde{b}_{2d}]$, utilisée pour générer des signatures.

On impose aussi que $\mathbf{B}_{f,g}$ soit *de bonne qualité* : $\max_i \|\widetilde{b}_i\|$ doit être petit.

(2) $\mathtt{Sign}(\mathbf{B}_{f,g}, \widetilde{\mathbf{B}}_{f,g}, M, \sigma, \beta)$ fait les opérations suivantes

- On hash le message par $\mathtt{H}(M) = m$ dans R_q et on note $c = (0, -m)$ une pré-image pour m par $\mathbf{A}$.
- On calcule $v = \mathtt{Sample}(\mathbf{B}_{f,g}, \widetilde{\mathbf{B}}_{f,g}, c, \sigma)$. Par la Proposition 6.22, en prenant $\sigma \geqslant \max_i \|\widetilde{b}_i\| \cdot \eta_\varepsilon(\mathbb{Z})$, v suit essentiellement une distribution Gaussienne $D_{\mathscr{L}_{\mathrm{NTRU}},c,\sigma}$. Etant essentiellement Gaussienne, cette distribution se concentre fortement vers sa moyenne : $\|v - c\| \leqslant 2 \cdot \sigma\sqrt{2d}$ avec très forte probabilité. Quand c'est le cas, on renvoie $(M, s = v - c)$.

(3) $\mathtt{Verif}(h, M, s, \beta)$ utilise $\beta = 2\sigma\sqrt{2d}$ comme paramètre public. La procédure vérifie que $[h, -1] \cdot s \equiv \mathtt{H}(M) \bmod q$ et que $\|s\| \leqslant \beta$. Si ces deux tests passent, (M,s) est valide.

Exercice 7.2. Une signature est un vecteur $s = (s_1, s_2) \in \mathbb{Z}^{2d}$, correspondant à $s_1, s_2 \in R_q$. On peut modifier `Verif` de deux façons pour limiter la quantité de données qui transitent, tout en conservant la vérification que le hash correspond à s et que s est assez courte. Si h est donné publiquement, montrer qu'envoyer s_2 est suffisant pour faire une vérification. Si h n'est pas donné, montrer qu'avoir (s_1, s_2) avec s_2 inversible modulo q est suffisant.

En terme d'efficacité, l'Algorithme 5 derrière `Sample` demande un nombre quadratique d'appel à un échantillonneur d'entiers Gaussiens, et ne peut être parallélisé — ceci fait une vraie différence de performance en pratique, d'autant plus qu'il faut stocker de quoi utiliser la Gram-Schmidt. La dernière brique pour FALCON est une version récursive quasi-linéaire de cet algorithme lorsque le réseau sous-jacent est un réseau NTRU, exploitant la tour de corps de nombres cyclotomiques sous-jacente à la manière de l'algorithme de transformée de Fourier rapide de Cooley et Tuckey. Présenter l'algorithme dans sa totalité n'est pas l'objectif de ces notes, les lecteurs intéressés peuvent lire la documentation de FALCON ou [DP16].
Les derniers obstacles sont les suivants, et concernent la génération de trappes [30] :

— générer $\mathbf{B}_{f,g}$ composée de petits vecteurs ;

— s'assurer que $\max \|\widetilde{b_i}\|$ est petit aussi.

7.6. **Générer une base de vecteurs courts de $\mathscr{L}_{\mathbf{NTRU}}$.** Le choix de f, g étant libre, il est normal de commencer par choisir $f, g \in R$ pour que $\|f, g\| \approx \sqrt{q}$ pour garantir que la moitié de la base est courte. En effet, les $d-1$ colonnes suivantes de $\mathbf{B}_{f,g}$ correspondent aux vecteurs $(\zeta^i f, \zeta^i g)$, et comme la matrice $[\zeta^i]$ est orthogonale (pourquoi ?), ils ont la même norme que (f, g).

La preuve du Lemme 7.3 ne dit pas si un vecteur (F, G) existe toujours ni comment le trouver le cas échéant, alors l'énoncé ci-dessous se charge d'y remédier. Avec un peu de théorie algébrique des nombres, il est possible de le rendre plus élégant [31], mais en l'état il sera suffisant pour ces notes (les choses belles seront dans l'annexe future).

Lemme 7.3. Si $\det[f]$ et $\det[g]$ sont premiers entre eux, alors on peut calculer $(F, G) \in R^2$ tel que $fG - gF = q$ en un nombre au plus cubique d'opérations.

Démonstration. Par le théorème de Bézout, il existe des entiers u, v tels que $(qu)\det[f] + (qv)\det[g] = q$. Le réseau $\mathscr{L}([f])$ est de déterminant $\det[f]$ et contenu dans $\mathbb{Z}^d$, autrement dit $|\mathbb{Z}^d / \mathscr{L}([f])| = \det[f]$. Ceci implique (pourquoi ?) que $\det[f] \in (f) \cap \mathbb{Z}$, où (f) est l'idéal principal engendré par $f \in R$. Il existe donc $\alpha \in R$ tel que $\det[f] = \alpha f$, qu'on peut calculer en prenant la première colonne de $[\alpha] = \det[f] \cdot [f]^{-1}$. L'inversion naïve est cubique en la taille des matrices, et c'est cette étape qui domine. On trouve de manière identique $\beta \in R$ tel que $\det[g] = \beta g$. Le résultat annoncé vient en posant $G = qu\alpha$ et $F = -qv\beta$ dans l'équation de Bézout. □

Les matrices en jeu sont de grande dimension, donc un examen rapide de la preuve montre que les nombres impliqués vont être généralement très gros, même si f, g ont des petits coefficients (l'inégalité de Hadamard (Corollaire 2.5) donne par exemple $\log(\det[f]) \leqslant d \log \|\mathbf{f}\|$, et en pratique $d \geqslant 512$). C'est ennuyeux pour les performances, mais de

30. C'est un problème généralement difficile qui intéresse beaucoup les cryptologues.

31. On peut simplement demander que les idéaux principaux $(f), (g)$ vérifient $(f) + (g) = R$.

toute façon en pratique, ce n'est pas cet algorithme qui est utilisé. On lui préfère une version récursive encore une fois [PP19], reposant sur la structure de tour de corps de nombres présente, et de complexité quasi-linéaire.

De manière bien plus problématique, cela implique que F, G obtenus par le Lemme 7.3 vont vraisemblablement avoir des gros coefficients aussi. On peut utiliser ici la structure algébrique de K-espace vectoriel ambiant et une intuition géométrique [32] pour améliorer considérablement la situation. Moralement, on a affaire à un objet 2-dimensionnel, et on nous donne un petit vecteur (f,g), et un gros vecteur $(\bar{F},\bar{G})$. Retrancher un R-multiple de (f,g) à $(\bar{F},\bar{G})$ ne sort pas du module NTRU, et on aimerait donc ici faire une étape de size-réduction (voir Section 3.1). Sur $\mathbb{Z}$, ce n'est pas un problème, mais ici on travaille sur un corps de nombres, donc il faut donner du sens à cette étape. Comme K est un corps *cyclotomique*, il possède un bon analogue [33] de produit scalaire comme on le verra dans la prochaine section.

La deuxième subtilité est que, sur $\mathbb{Q}$, un μ n'a qu'un seul entier le plus proche, mais sur K, il n'y a pas qu'une façon unique de prendre un proche entier algébrique. En pratique, on se simplifie la vie, et on prend le polynôme $\lfloor \mu \rceil = \sum_i \lfloor \mu_i \rceil \zeta^i$ — autrement dit, on fait Round-off.

D'un point de vue géométrique, le deuxième vecteur (F,G) après la size-réduction sur K est proche d'être orthogonal à (f,g). Sa longueur va donc fortement dépendre de q et de celle de (f,g) comme dans un rectangle. Comme on a choisit $\|f,g\| \approx \sqrt{q}$, on devrait aussi avoir $\|F,G\| \approx \sqrt{q}$.

7.7. **Qualité de la base pour l'échantillonnage Gaussien.** D'après la Proposition 6.22, l'écart-type que `Sample` peut atteindre est $\sigma = \max_i \|\widetilde{b}_i\| \cdot \eta_\varepsilon(\mathbb{Z})$, où les $\widetilde{b}_i$ sont les colonnes d'une $\widetilde{\mathbf{B}}_{f,g}$. Pour la suite, on omettra le facteur $\eta_\varepsilon(\mathbb{Z})$ qui est une constante publique proche de 1. La structure algébrique sous-jacente va être cruciale, à la fois pour estimer finement la valeur de σ, mais aussi pour l'efficacité de `KeyGen`. Elle va servir à montrer le résultat suivant.

Proposition 7.4. Pour $1 \leqslant i \leqslant d$, soit $b_i = (\zeta^i f, \zeta^i g), b_{d+i} = (\zeta^i F, \zeta^i G)$ les vecteurs d'une base $\mathbf{B}_{f,g}$ d'un réseau NTRU. Si $(\widetilde{b}_i)_i$ désigne son orthogonalisée de Gram-Schmidt, on a

$$\max_i \|\widetilde{b}_i\| = \max(b_1, \widetilde{b}_{d+1}).$$

Pour les d premiers vecteurs, comme chaque $\widetilde{b}_i$ est une projection du $b_i = (\zeta^i f, \zeta^i g)$ correspondant, on a $\|b_1\| = \|b_i\|$ pour $1 \leqslant i \leqslant d$ et il vient :

$$\|(f,g)\| = \max_{1 \leqslant i \leqslant d} \|\widetilde{b}_i\|.$$

Pour la deuxième partie de la base, il faut un peu plus de travail. Le but va être de montrer que le $d+1$-ème Gram-Schmidt est forcément le plus grand des d derniers.

Gram-Schmidt sur un corps cyclotomique. Ainsi, on a $\bar{\zeta} = \zeta^{-1} = -\zeta^{d-1}$ ce qui implique que la conjugaison complexe est un automorphisme de K. Si $f = \sum f_i \zeta^i$, on définit alors $f^* = \sum_i f_i \bar{\zeta}^i = f_0 - \sum_i f_{d-i} \zeta^i$. On observe de plus que cet élément correspond à la première *ligne* de $[f]$, donc que $[f^*] = [f]^t$. D'autre part, ceci permet de définir une forme bilinéaire "hermitienne" sur K^2 par

$$\langle (f,g), (F,G) \rangle_K = f^* F + g^* G.$$

32. Les dessins viendront, mais c'est bien de les faire soi-même aussi.

33. En tant que corps à multiplication complexe (*CM fields*), les corps cyclotomiques ont en particulier des propriétés structurelles similaires à la situation de $\mathbb{C}$ par rapport à $\mathbb{R}$.

Deux vecteurs de K^2 sont K-orthogonaux si la forme ci-dessus s'y évalue à 0. Notons que la représentation matricielle de $\langle (f,g),(f,g)\rangle_K = ff^* + gg^*$ est $[f]^t[f] + [g]^t[g]$, qui est une matrice symétrique définie positive (pourquoi ?) et donc toujours inversible. En d'autres termes, un élément de la forme $ff^* + gg^*$ n'est nul que si f,g sont nuls. Etant donnés deux vecteurs $b_1 = (f,g), b_2 = (F,G) \in K^2$, on peut étendre le procédé de Gram-Schmidt naturellement :

$$\widetilde{b}_1^K = b_1 \quad \text{et} \quad \widetilde{b}_2^K = b_2 - \frac{\langle b_1, b_2\rangle_K}{\langle b_1, b_1\rangle_K} \cdot b_1,$$

et il n'est pas surprenant que $\widetilde{b}_1^K$ et $\widetilde{b}_2^K$ soient K-orthogonaux. Le résultat suivant permet de lier l'orthogonalisation de Gram-Schmidt sur $\mathbb{Q}$ et sur K ; bien qu'elles ne soient pas équivalente, la seconde peut être vue comme une version "par blocs" de la première.

Proposition 7.5. si $(f,g),(F,G)$ correspondent aux colonnes d'une base $\mathbf{B}_{f,g}$ d'un réseau NTRU, on a

$$\widetilde{b}_2^K := \begin{bmatrix} \widetilde{F} \\ \widetilde{G} \end{bmatrix} = \begin{bmatrix} \frac{-qg^*}{ff^*+gg^*} \\ \frac{qf^*}{ff^*+gg^*} \end{bmatrix}.$$

De plus, si $b_{d+1} = (F,G)$ dans $\mathbb{Q}$, alors $\widetilde{b}_{d+1} = \widetilde{b}_2^K$.

Démonstration. L'expression de $\widetilde{b}_2^K$ est un calcul laissé au lecteur, en utilisant le fait que $fG - gF = q$. On note ensuite π la projection K-orthogonale telle que $\pi(b_2) = \frac{\langle b_1,b_2\rangle_K}{\langle b_1,b_1\rangle_K} \cdot b_1$. On peut l'identifier à la projection orthogonale P (dans $\mathbb{R}^{2d}$) sur l'espace engendré par les d premières colonnes de $\mathbf{B}_{f,g}$. En effet, on peut écrire P comme la matrice

$$P = \begin{bmatrix} [f] \\ [g] \end{bmatrix} \cdot ([f]^t[f] + [g]^t[g])^{-1} \cdot ([f]^t, [g]^t) = \begin{bmatrix} [f] \\ [g] \end{bmatrix} \cdot \left(\begin{bmatrix} \frac{f^*}{ff^*+gg^*} \end{bmatrix}, \begin{bmatrix} \frac{g^*}{ff^*+gg^*} \end{bmatrix} \right),$$

ce qui donne immédiatement l'identification. La première colonne de $[\widetilde{b}_2^K] = [b_2 - \pi(b_2)]$ est donc bien la projection orthogonale de (F,G) sur l'orthogonal de l'espace engendré par les d premières colonnes de $\mathbf{B}_{f,g}$. □

La preuve ci-dessus montre que $[\widetilde{b}_2^K]$ est composé des vecteurs $(\widetilde{F},\widetilde{G}),(\zeta\widetilde{F},\zeta\widetilde{G}),\ldots,(\zeta^{d-1}\widetilde{F},\zeta^{d-1}\widetilde{G})$, qui ont donc tous la même norme. Avec le même raisonnement qu'en début de section, on en déduit $\|\widetilde{b}_{d+1}\| = \max_{d+1\leqslant i\leqslant 2d} \|\widetilde{b}_i\|$, et le résultat annoncé en Proposition 7.4.

7.8. **Finaliser `KeyGen`.** Notons désormais $Q(f,g) = \max(b_1, \widetilde{b}_{d+1})/\sqrt{q}$. Les sections précédentes ont grandement simplifié la situation, et en pratique on cherche maintenant des bases telle que $Q(f,g) \approx 1$. Plus $Q(f,g)$ est grand, plus il est facile de trouver une bonne paire, mais bien sûr la sécurité du schéma diminue. En pratique, on essaie d'atteindre $Q(f,g) = 1.17$, une valeur correspondant à un niveau de sécurité équivalent à la difficulté de casser $\texttt{AES}_{128}$. Elle est déterminée à la fois heuristiquement et expérimentalement, voir [Pre15].

Il reste un dernier problème : il n'y a aucune raison pour qu'une paire (f,g) de taille $\approx \sqrt{q}$ donne une base de bonne qualité pour l'échantillonnage. Malheureusement il n'y a pas grand chose à espérer de plus que de tenter sa chance en tirant au hasard dans tout l'espace de paires (f,g) de cette taille. Au final, `KeyGen` ressemble au pseudo-code ci-dessous.

Ici, on utilise *crucialement* la Proposition 7.5, qui montre que $\widetilde{b}_2$ est complètement déterminé par f,g et q. En pratique, cela veut dire qu'on peut savoir si une base $\mathbf{B}_{f,g}$ va avoir une bonne qualité *sans avoir besoin* de la compléter

Algorithm 6: `KeyGen` simplifié

Tirer $f,g \leftarrow D_{\mathbb{Z}^d,\sqrt{\frac{q}{2d}}}$ jusqu'à ce que f soit inversible $\bmod q$ et $\det[f], \det[g]$ soient premiers entre eux;

si $Q(f,g) \leqslant 1.17$ **alors**

  Compléter la base $\mathbf{B}_{f,g} = [(f,g),(F,G)]$;

  $h = g/f \bmod q$;

  Calculer $\widetilde{\mathbf{B}}_{f,g}$;

  Renvoyer $(\mathtt{pk} = h, \mathtt{sk} = (\mathbf{B}_{f,g}, \widetilde{\mathbf{B}}_{f,g}))$;

fin si

sinon

  recommencer;

fin si

avant, c'est-à-dire, sans avoir besoin de connaitre (F,G). Comme le temps d'exécution de l'algorithme de complétion de base domine très largement le reste, même en utilisant la variante quasi-linéaire, on est bien content de n'avoir à le faire qu'une seule fois.

Le choix de prendre f,g Gaussien est un choix confortable : avec forte probabilité, on aura $\|f,g\| \approx \sqrt{q}$, et on sait plutôt bien échantillonner sur $\mathbb{Z}$. Les conditions de coprimalité sont vérifiés la plupart du temps, mais il n'est pas si simple de le démontrer (voir par exemple [SS11]).

7.9. **Paramétrage et dernières remarques.** D'après la section précédente, `KeyGen` renvoie une base $\mathbf{B}_{f,g}$ telle que $Q(f,g) = 1.17$. En d'autres termes, on peut choisir $\sigma \approx 1.17\sqrt{q}$, et on s'attend à ce que les signatures générées soient de longueur très proches de $\sigma\sqrt{2d}$. On fixe une borne d'acceptation β très légèrement plus grande que cette quantité pour augmenter le taux du succès de génération des signatures.Les dimensions considérées en pratique sont $d = 512$ ou $d = 1024$, ce qui ne donne pas beaucoup de flexibilité. Le modulus q est choisi spécifiquement à $q = 12289$ pour ses propriétés arithmétiques qui permettent des multiplications efficaces (par Number Theoretic Transform (NTT)), particulièrement dans la procédure `Verif`.

En pratique, l'algorithme est sensiblement plus technique que ce que j'ai décrit, à cause des algorithmes récursifs de type Fast Fourier Transform (FFT) pour `KeyGen` et `Sample`. Par ailleurs, la FFT est une partie intégrante de `KeyGen`, et force l'emploi de nombre en virgule flottante et d'analyse de précision puisqu'il faut multiplier ou diviser avec des réels (qu'on ne sait pas représenter en machine).

Enfin, le résultat décrit dans l'Exercice 7.2 est utilisé en pratique : on n'envoie que la moitié d'un vecteur signature, ce qui réduit la consommation en bande passante. En fait, on n'envoie même pas le vecteur lui-même, mais un encodage qui permet de réduire encore la taille des données en transit. Au final, les signatures FALCON font 666 Bytes pour $d = 512$ et 1280 Bytes pour $d = 1024$. Pour le premier, c'est à peu près trois fois plus gros que les actuelles signatures RSA, par contre l'implémentation (non optimisée !) est déjà cinq fois plus rapide que celle de ces mêmes signatures RSA. Par rapport aux signatures de courbes elliptiques de 64 Bytes, il y a un ordre de grandeur de différence qui va être difficile à combler à cause de la nature des données employées.

RÉFÉRENCES

[CPS+20] Chitchanok Chuengsatiansup, Thomas Prest, Damien Stehlé, Alexandre Wallet, and Keita Xagawa. ModFalcon : Compact signatures based on module-NTRU lattices. In *ASIACCS 2020*, pages 853–866, 2020.

[CS88] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Grundlehren der Mathematischen Wissenschaften 290. Springer-Verlag, New York, 1988.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41. Springer, Heidelberg, December 2014.

[DN12] Léo Ducas and Phong Q. Nguyen. Learning a zonotope and more : Cryptanalysis of NTRUSign countermeasures. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 433–450. Springer, Heidelberg, December 2012.

[DP16] Léo Ducas and Thomas Prest. Fast Fourier Orthogonalization. In *ISSAC 2016*, pages 191–198, 2016.

[EFG+22] Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Mitaka : A simpler, parallelizable, maskable variant of falcon. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 222–253. Springer, Heidelberg, May / June 2022.

[ETWY22] Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Shorter hash-and-sign lattice-based signatures. In *Crypto 2022*, 2022.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[HHP+03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN : Digital signatures using the NTRU lattice. In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 122–140. Springer, Heidelberg, April 2003.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1) :267–302, 2007.

[NR06] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped : Cryptanalysis of GGH and NTRU signatures. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 271–288. Springer, Heidelberg, May / June 2006.

[PP19] Thomas Pornin and Thomas Prest. More efficient algorithms for the NTRU key generation using the field norm. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 504–533. Springer, Heidelberg, April 2019.

[Pre15] Thomas Prest. *Gaussian Sampling in Lattice-Based Cryptography*. PhD thesis, École Normale Supérieure, Paris, France, 2015.

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, Heidelberg, May 2011.