Université Paris 8 J. Lavauzelle

Master 2 Mathématiques et applications — année 2023–24

Algorithmes Arithmétiques II – Devoir à la maison

Transmis le 07/11/2023 à rendre pour le **31/12/2023**

Documents à fournir. Vous devez rendre, par email adressé `julien.lavauzelle@univ-paris8.fr` et jusqu'au **31 décembre 2023** dernier délai, une archive au format `.zip` contenant les documents ci-dessous :

1. un fichier édité sur ordinateur (pas de photos), au format `.pdf`, contenant vos réponses aux questions théoriques (l'utilisation de LaTeX est fortement conseillée) ;
2. vos fichiers de programmation pour les questions d'implantation.

Vous pouvez utiliser le langage de programmation de votre choix. Les logiciels de calcul formel comme `sage` ou `magma` sont conseillés, mais la plupart des questions peuvent être réalisées dans des langages basiques (`python` ou `C/C++` par exemple).

Le **soin** et les **justifications** apportées à vos réponses seront évaluées. On notera également le soin apporté à l'implantation (commentaires, lisibilité, etc.).

Pour se documenter, on pourra se référer aux ressources suivantes :

[LLL82] H.W. Lenstra, A.K. Lenstra, and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261 :515–534, 1982

[BCG+17, Chapitre 20] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. *Algorithmes Efficaces en Calcul Formel*. Frédéric Chyzak (auto-édit.), 2017

[vzGG13, Chapitre 16] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra (3. ed.)*. Cambridge University Press, 2013

Une introduction à la réduction dans les réseaux euclidiens

Présentation du sujet

Le sujet a pour objectif de découvrir les aspects algorithmiques de la **réduction de bases de réseaux euclidiens**.

Pour rappel, de nombreux problèmes algorithmiques sur les réseaux euclidiens sont à la base de la sécurité de systèmes de chiffrement et de signature post-quantiques, comme par exemple :

- *les problèmes* CVP *et* SVP *(*closest *et* shortest vector problem*) qui demandent de calculer un vecteur le plus proche, ou le plus court, dans un réseau ;*
- *le problème* LWE *(*learning with errors*), qui demande de retrouver une base d'un rseau à partir de vecteurs bruités ;*
- *le problème* NTRU *(*Nth truncated ring units*), qui est une variation d'un problème de recherche de vecteur court.*

Organisation. Le sujet est découpé en trois parties :

1. une introduction qui permettra de définir les notions et les problémantiques essentielles du domaine ;
2. une partie qui concerne la réduction optimale de bases en dimension 2 ;
3. une partie qui concerne la réduction effective de bases en dimension supérieure (algorithme LLL).

Évaluation. Le sujet est assez long et contient quelques questions avancées. Il n'est pas attendu de traiter la totalité les questions pour obtenir la note maximale. En revanche, une attention particulière sera portée à la **qualité et au soin des raisonnements et des implémantations**. Il est préférable de traiter excellement la moitié des questions que moyennement toutes les questions.

1 Introduction

Notations.

- on note $\lfloor x \rceil$ l'arrondi entier de $x \in \mathbb{R}$, avec comme convention $\lfloor k + \frac{1}{2} \rceil = k + 1$ pour tout $k \in \mathbb{Z}$
- $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \coloneqq \sum_{i=1}^{n} u_i v_i$ est le produit scalaire euclidien entre deux vecteurs $\boldsymbol{u}, \boldsymbol{v}$ de $\mathbb{R}^n$
- $\|\boldsymbol{u}\| \coloneqq \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle}$ est la norme euclidienne du vecteur $\boldsymbol{u} \in \mathbb{R}^n$
- $\mathcal{B}_\epsilon(\boldsymbol{v}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{v} - \boldsymbol{x}\| < \epsilon\}$ est la boule ouverte de centre $\boldsymbol{v} \in \mathbb{R}^n$ et de rayon $\epsilon \in \mathbb{R}^+$
- les vecteurs sont usuellement représentés par des colonnes, et on assimile une séquence de vecteurs avec la matrice dont les colonnes sont ces vecteurs
- l'ensemble des matrices de taille $m \times n$ sur un anneau A est noté $A^{m \times n}$
- on note $\mathrm{GL}_n(A)$ l'ensemble des matrices de taille $n \times n$ inversibles sur un anneau A
- le déterminant d'une matrice $\boldsymbol{M}$ est noté $\det(\boldsymbol{M})$
- l'espace vectoriel engendré par des vecteurs $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$ est noté $\mathrm{Vect}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$
- si $\mathcal{V}$ est un sous-espace vectoriel de $\mathbb{R}^n$, alors $\mathcal{V}^\perp$ représente son orthogonal

1.1 Définitions et premiers résultats

Dans cette première partie, nous donnons les définitions principales et les premiers résultats élémentaires concernant les réseaux euclidiens.

Définition 1.1

Un réseau euclidien *(ou simplement, un* réseau*) est un sous-groupe $\mathcal{L}$ de $(\mathbb{R}^n, +)$ qui est discret :*

$$\forall \boldsymbol{v} \in \mathcal{L}, \quad \exists \epsilon > 0, \quad \mathcal{B}_\epsilon(\boldsymbol{v}) \cap \mathcal{L} = \{\boldsymbol{v}\}.$$

Par exemple, le sous-groupe $\mathbb{Z}^n$ est un réseau : tous ses vecteurs sont distants d'au moins 1. En revanche, le sous-groupe $\mathbb{Q}^n$ n'est pas un réseau, car il n'est pas discret. La Figure 1 représente deux réseaux euclidiens du plan.

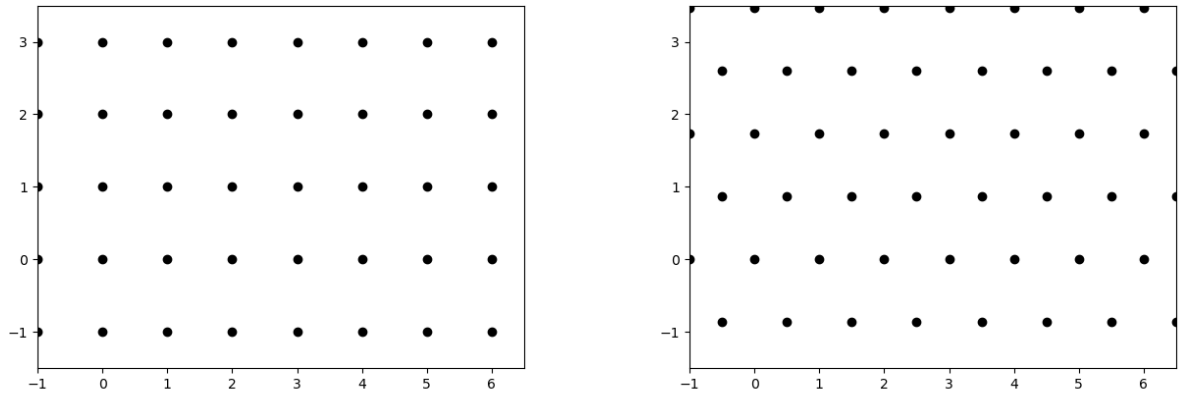

FIGURE 1 – Illustration de deux réseaux en dimension 2 : à gauche le réseau $\mathbb{Z}^2$, à droite le réseau *hexagonal*.

Une définition plus explicite et calculatoire d'un réseau est la suivante.

Définition 1.2

Soit $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m) \in (\mathbb{R}^n)^m$ une famille de m vecteurs linéairement indépendants. Le réseau engendré par $\boldsymbol{A}$ est :

$$\mathcal{L}(\boldsymbol{A}) \coloneqq \{\lambda_1 \boldsymbol{a}_1 + \cdots + \lambda_m \boldsymbol{a}_m \mid (\lambda_1, \ldots, \lambda_m) \in \mathbb{Z}^m\}.$$

La famille $\boldsymbol{A}$ est appelée base *du réseau, et on confondra souvent la famille $\boldsymbol{A}$ avec la matrice dont les colonnes sont les éléments de la famille.*
L'entier n est appelé la dimension *du réseau, et l'entier m son* rang. *Si $n = m$, on dit que le réseau est de rang plein.*

Par exemple, le réseau canonique $\mathbb{Z}^n$ est engendré par les vecteurs de la base canonique $\boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$, où $1 \leq i \leq n$.

Remarque 1.3

Tout réseau $\mathcal{L}$ est engendré par une base $\boldsymbol{A}$. Autrement dit, tout réseau $\mathcal{L}$ peut être construit en appliquant une transformation linéaire (pas forcément injective) sur le réseau canonique $\mathbb{Z}^n$.

Question 1.– Trouver deux vecteurs non-nuls et distincts $\boldsymbol{u}$ et $\boldsymbol{v}$ dans $\mathbb{R}^2$, tels que la famille $(\boldsymbol{u}, \boldsymbol{v})$ n'engendre pas un réseau euclidien.

Il n'y a pas unicité de la base d'un réseau, et on peut même caractériser l'ensemble de ses bases par le résultat suivant.

Proposition 1.4

Soient $\boldsymbol{A}$ et $\boldsymbol{B}$ deux matrices de $\mathbb{R}^{n\times m}$ de rang m qui engendrent le même réseau $\mathcal{L} = \mathcal{L}(\boldsymbol{A}) = \mathcal{L}(\boldsymbol{B})$. Alors, il existe une matrice unimodulaire $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$ telle que $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{B}$.

Rappelons que toute matrice entière unimodulaire a déterminant ± 1, et qu'elle s'écrit comme produit de matrices élémentaires dont l'action à gauche donne les opérations suivantes :

1. échange de deux colonnes,
2. multiplication d'une colonne par -1,
3. ajout à une colonne d'une multiple entier d'une autre colonne.

Définissons maintenant deux grandeurs importantes d'un réseau.

Définition 1.5

Le déterminant *d'un réseau $\mathcal{L}$, noté* $\det(\mathcal{L})$, *est la valeur absolue du déterminant de n'importe quelle base du réseau.*
Le domaine élémentaire *associé à une base $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m)$ d'un réseau, est l'ensemble*

$$\mathcal{F}(\boldsymbol{a}) := \{t_1\boldsymbol{a}_1 + \cdots + t_m\boldsymbol{a}_m \mid (t_1, \ldots, t_m) \in [0,1]^m\}.$$

Notons que le volume de n'importe quel domaine élémentaire d'un réseau est égal au déterminant du réseau. Par ailleurs, le domaine élémentaire d'un réseau permet de paver de sous-espace vectoriel qu'il engendre, comme l'atteste la Figure 2.

Parmi les différentes bases d'un réseau, certaines seront plus aptes à effectuer des calculs. Ce sont les bases dont les vecteurs sont les plus courts et plus « proches » d'être deux à deux orthogonaux. Ces bases seront appelées informellement « bonnes bases ». Dans la Figure 3, on représente une bonne base et une mauvaise base d'un même réseau.

Par ailleurs, pour certaines applications (par exemple en cryptographie), il peut être utile de savoir calculer des vecteurs relativement courts d'un réseau. En particulier, on définit les objets suivants.

Définition 1.6

Soit $\mathcal{L}$ un réseau de $\mathbb{R}^n$. On note :
— $\lambda_1(\mathcal{L}) := \min\{\|\boldsymbol{v}\|, \boldsymbol{v} \in \mathcal{L} \setminus \{\boldsymbol{0}\}\}$ *la plus courte norme d'un vecteur (non-nul) du réseau ;*
— $\Lambda_1(\mathcal{L}) := \{\boldsymbol{v} \in \mathcal{L} \mid \|\boldsymbol{v}\| = \lambda_1(\mathcal{L})\}$ *l'ensemble des vecteurs (non-nuls) les plus courts d'un réseau.*

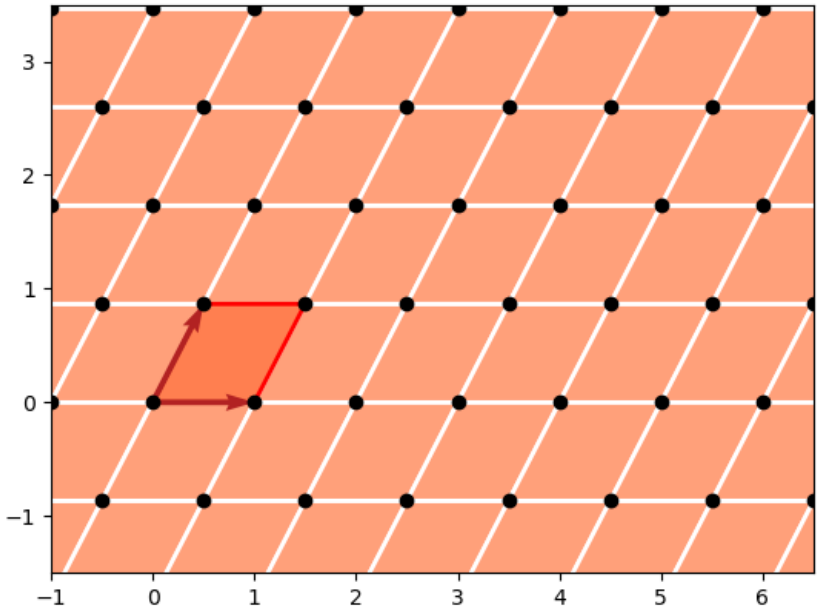

FIGURE 2 – Domaine élémentaire et pavage du plan

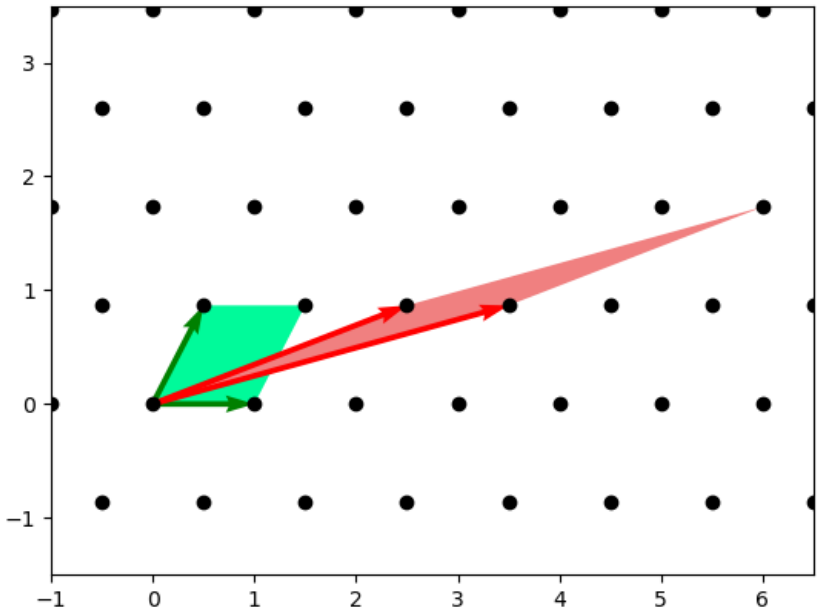

FIGURE 3 – Bonne (en vert) et mauvaise (en rouge) bases d'un même réseau, ainsi que leurs domaines fondamentaux associés. L'aire de ces deux domaines est identique.

Problématique. L'objectif de ce devoir est de présenter une méthode algorithmique qui permet de transformer une base quelconque d'un réseau en une base relativement bonne. Pour cela, nous avons besoin de comprendre comment résoudre ce problème dans les espaces vectoriels.

1.2 Rappel : orthogonalisation de Gram–Schmidt

Dans le cas d'**espaces vectoriels**, une manière de transformer une base quelconque de l'espace en une base « adaptée au calcul » est de suivre le procédé d'orthogonalisation (ou parfois même d'orthonormalisation) de Gram–Schmidt.

L'Algorithme 1 rappelle cette méthode.

Algorithme 1 : Procédé d'orthogonalisation de Gram–Schmidt

Entrée : une base $(\boldsymbol{a}_1, \dots, \boldsymbol{a}_m)$ d'un espace vectoriel $\mathcal{A}$
Sortie : une base orthononale $(\boldsymbol{a}_1^*, \dots, \boldsymbol{a}_m^*)$ de $\mathcal{A}$, telle que $\mathrm{Vect}(\boldsymbol{a}_1, \dots, \boldsymbol{a}_k) = \mathrm{Vect}(\boldsymbol{a}_1^*, \dots, \boldsymbol{a}_k^*)$ pour tout $1 \leq k \leq m$

```
Définir a_1^* ← a_1
Pour tout i = 2, ..., m faire
    Pour tout j = 1, ..., i − 1 faire
        Calculer μ_{i,j} ← ⟨a_i, a_j^*⟩ / ‖a_j^*‖²
    Définir a_i^* ← a_i − Σ_{j=1}^{i−1} μ_{i,j} a_j^*
Retourner (a_1^*, ..., a_m^*)
```

On note que dans le procédé de Gram-Schmidt, le vecteur $\sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{a}_j^*$ est le projeté orthogonal de $\boldsymbol{a}_i$ sur le sous-espace $\mathrm{Vect}(\boldsymbol{a}_1^*, \dots, \boldsymbol{a}_{i-1}^*)$.

Question 2.– Quelle est la complexité (en nombre d'opérations élémentaires sur un scalaire) du procédé d'orthogonalisation de Gram–Schmidt ?

Question 3.– Implanter le procédé d'orthogonalisation de Gram–Schmidt. Puis, tester votre fonction sur les familles suivantes :

1. en dimension 2 : $(\boldsymbol{u}, \boldsymbol{v})$ avec $\boldsymbol{u} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$ et $\boldsymbol{v} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$
2. en dimension 3 : $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ avec $\boldsymbol{a} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, $\boldsymbol{b} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$, et $\boldsymbol{b} = \begin{pmatrix} 0 \\ -3 \\ 0 \end{pmatrix}$
3. en dimension 5 avec les colonnes de la matrice :

$$\boldsymbol{M} = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 1 & -1 & 4 & 5 \\ 1 & -2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 3 & -1 \end{pmatrix}$$

Question 4.– Pourquoi ne peut-on pas directement appliquer le procédé d'orthogonalisation de Gram–Schmidt pour obtenir une bonne base d'un réseau euclidien ?

2 Réduction en dimension 2

En dimension 2, il existe une méthode efficace pour obtenir une base d'un réseau contenant :

— un des vecteurs les plus courts du réseau,
— un autre vecteur, dont l'angle avec le premier vecteur est compris en valeur absolue entre $\pi/3$ et $2\pi/3$.

Cette méthode est appelée *algorithme de Gauss* (parfois algorithme de Lagrange–Gauss) et nous la décrivons ci-dessous dans l'Algorithme 2.

Algorithme 2 : Algorithme de Gauss pour la réduction de réseaux en dimension 2

Entrée : une base $(\boldsymbol{a}, \boldsymbol{b})$ d'un réseau $\mathcal{L} \subseteq \mathbb{R}^2$, avec $\|\boldsymbol{a}\| \leq \|\boldsymbol{b}\|$
Sortie : une base minimale du réseau $\mathcal{L}$

1 Calculer $m = \left\lfloor \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\|^2} \right\rceil$.
2 **Tant que** $m \neq 0$ **faire**
3 Remplacer $\boldsymbol{b}$ par $\boldsymbol{b} - m\boldsymbol{a}$.
4 Échanger $\boldsymbol{a}$ et $\boldsymbol{b}$.
5 Calculer $m = \left\lfloor \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\|^2} \right\rceil$.
6 **Retourner** $(\boldsymbol{b}, \boldsymbol{a})$.

En quelques mots, l'idée de la méthode de Gauss est de chercher, itérativement, le vecteur le plus proche du vecteur que le procédé de Gram–Schmidt produirait. Pour cela, on arrondit vers l'entier le plus proche le coefficient μ calculé lors de l'orthogonalisation de Gram–Schmidt.

Question 5.– On souhaite démontrer que l'algorithme de Gauss retourne, dans sa deuxième entrée, un des vecteurs les plus courts du réseau $\mathcal{L}$ engendré par $\boldsymbol{a}$ et $\boldsymbol{b}$. Pour éviter les confusions, notons $(\boldsymbol{u}, \boldsymbol{v})$ la sortie de l'algorithme.

1. Justifier pourquoi la sortie de l'algorithme forme bien une base du réseau $\mathcal{L}(\boldsymbol{a}, \boldsymbol{b})$.
2. Démontrer qu'en sortie de l'algorithme, on a
$$\frac{|\langle \boldsymbol{u}, \boldsymbol{v} \rangle|}{\|\boldsymbol{v}\|^2} \leq \frac{1}{2}.$$
3. Soit $\boldsymbol{w} = x\boldsymbol{u} + y\boldsymbol{v} \in \mathcal{L}$, avec $(x, y) \in \mathbb{Z}^2$. Démontrer que
$$\|\boldsymbol{w}\|^2 \geq (x^2 - |xy| + y^2)\|\boldsymbol{v}\|^2$$
4. En déduire que le vecteur $\boldsymbol{v}$ est dans $\Lambda_1(\mathcal{L}(\boldsymbol{a}, \boldsymbol{b}))$.

Au passage, on a démontré avec le premier point de la question précédente, que l'angle entre les vecteurs de sortie de l'algorithme de Gauss est en valeur absolue compris entre $\pi/3$ et $2\pi/3$: ces vecteurs sont donc, en un sens, « presque orthogonaux ».

Question 6.– Implanter l'algorithme de Gauss, puis le tester avec les valeurs suivantes :

1. $\boldsymbol{a}_1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$ et $\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$
2. $\boldsymbol{a}_2 = \begin{pmatrix} 0 \\ -12 \end{pmatrix}$ et $\boldsymbol{b}_2 = \begin{pmatrix} 27 \\ 0 \end{pmatrix}$
3. $\boldsymbol{a}_3 = \begin{pmatrix} 6513996 \\ 6393464 \end{pmatrix}$ et $\boldsymbol{b}_3 = \begin{pmatrix} 66586820 \\ 65354729 \end{pmatrix}$

Dans la question suivante , on cherche à déterminer quelques propriétés sur le nombre de tours de boucle effectués par l'algorithme de Gauss. On appelle « tour de boucle » un passage entre les lignes 2 et 5 de l'Algorithme 2.

Question 7.– Soit $\boldsymbol{b} = (0,1) \in \mathbb{R}^2$. Pour tout $n \geq 1$, on note

$$A_n \coloneqq \{\boldsymbol{a} \in \mathcal{B}_1(\mathbf{0}) \setminus \mathrm{Vect}(\boldsymbol{b}) \mid \text{l'algorithme de Gauss effectue } \geq n \text{ tours de boucle sur l'entrée } (\boldsymbol{a}, \boldsymbol{b})\}.$$

L'ensemble A_n représente l'ensemble des vecteurs $\boldsymbol{a}$ du disque unité, qui ne sont pas colinéaires avec $\boldsymbol{b}$, et pour lesquels l'Algorithme 2 d'entrée $(\boldsymbol{a}, \boldsymbol{b})$ effectue au moins n tours de boucle.

1. Déterminer l'ensemble A_1 (on pourra s'intéresser à son complémentaire dans $\mathcal{B}_1(\mathbf{0})$).
2. Trouver expérimentalement un élément de $A_3 \setminus A_4$, et donner les valeurs successives obtenues pour les valeurs de $\boldsymbol{a}$ dans l'algorithme.
3. Expérimentalement, donner une représentation graphique de A_n pour de petites valeurs de n.

3 Réduction en dimension supérieure

L'algorithme de Gauss ne fonctionne pas directement en dimension supérieure à 2, car à chaque étape il faudrait choisir arbitrairement un vecteur à projeter (mais lequel ?) et un sous-espace sur lequel projeter (lequel également ?). L'ordre des vecteurs et des sous-espaces choisis influe grandement sur la sortie éventuelle de l'algorithme, et certains choix ne permettent pas d'aboutir rapidement à des bases courtes.

En réalité, en dimension n, il n'existe pas d'algorithme polynomial en n qui retourne une base contenant un des vecteurs les plus courts du réseau. On doit donc se rabattre sur des algorithmes qui fourniront de « meilleures bases » que des bases quelconuques. En général, on qualifie ces bases de *réduites*.

Lenstra, Lenstra et Lovacs ont proposé en 1982 [LLL82] une forme de réduction, et un algorithme permettant d'obtenir de telles bases. Voici la définition des bases dites *LLL-réduites*.

Définition 3.1

Soit $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ une base d'un réseau $\mathcal{L} \subseteq \mathbb{R}^n$ de dimension n. On note $\boldsymbol{B}^* = (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*)$ son orthogonalisation de Gram–Schmidt, et $\mu_i,j = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\|\boldsymbol{b}_j^*\|^2}$ les scalaires obtenu lors de l'othogonalisation. Alors, la base $\mathcal{B}$ est dite LLL-réduite si les conditions suivantes sont vérifiées :

1. *condition de taille :*
$$|\mu_{i,j}| \leq \frac{1}{2} \quad \text{pour tous } 1 \leq j < i \leq n,$$
2. *condition de Lovasz :*
$$\|\boldsymbol{b}_i^* + \mu_{i,i-1}\boldsymbol{b}_{i-1}^*\|^2 \geq \frac{3}{4}\|\boldsymbol{b}_{i-1}^*\|^2, \quad \text{pour tous } 2 \leq i \leq n,$$

Observons que la condition de Lovasz est équivalente à :

$$\left\|\text{Projection de } \boldsymbol{b}_i \text{ sur } \mathrm{Vect}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-2})^\perp\right\|^2 \geq \frac{3}{4}\left\|\text{Projection de } \boldsymbol{b}_{i-1} \text{ sur } \mathrm{Vect}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-2})^\perp\right\|^2$$

Question 8.– Parmi les bases suivantes, lesquelles sont LLL-réduites ? Si besoin, on pourra s'aider de la fonction Gram–Schmidt implantée dans la section précédente.

1. La base formée des colonnes de la matrice

$$\boldsymbol{M}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 2 & -1 \\ 1 & 1 & 3 & -1 \\ 0 & 1 & 4 & 1 \end{pmatrix}$$

2. La base formée des colonnes de la matrice

$$\boldsymbol{M}_2 = \begin{pmatrix} 0 & 1 & 1 & -1 \\ -1 & 1 & 0 & 2 \\ 0 & 1 & -1 & -1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

3. La base formée des colonnes de la matrice de taille $(n \times n)$:

$$\boldsymbol{T}_n = \begin{pmatrix} 2^n & 0 & 0 & \dots & 0 \\ 2^{n-1} & 2^n & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 2 & & \ddots & \ddots & 0 \\ 1 & 2 & \dots & 2^{n-1} & 2^n \end{pmatrix}$$

Pour cette dernière base, vous pouvez répondre d'abord par des tests expérimentaux, puis essayer d'apport une réponse générale.

Theorème 3.2

Soit $\mathcal{L}$ un réseau de dimension n. Toute base LLL-réduite $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ de $\mathcal{L}$ a les propriétés suivantes :

1. *Si $(\boldsymbol{b}_1^*, \dots, \boldsymbol{b}_n^*)$ est l'orthogonalisée de Gram–Schmidt de $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$, alors la norme des vecteurs de la base du réseau est bornée en fonction de celles des vecteurs de l'orthogonalisée :*

$$\|\boldsymbol{b}_i\| \leq 2^{(j-1)/2} \left\|\boldsymbol{b}_j^*\right\|, \qquad \forall 1 \leq i < j \leq n\,.$$

2. *Le produit des normes des vecteurs donne une approximation (exponentielle) du déterminant :*

$$\prod_{i=1}^{n} \|\boldsymbol{b}_i\| \leq 2^{\frac{n(n-1)}{4}} \det(\mathcal{L})\,.$$

3. *Le vecteur $\boldsymbol{b}_1$ est relativement court :*

$$\|\boldsymbol{b}_1\| \leq 2^{(n-1)/2} \lambda_1(\mathcal{L})\,.$$

On va démontrer une partie des résultats du Théorème 3.2 dans la question suivante :

Question 9.– Soit $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ une base LLL-réduite d'un réseau $\mathcal{L}$ et $(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*)$ son orthogonalisée de Gram–Schmidt.

1. En utilisant les conditions d'une base LLL-réduite, démontrer que

$$\|\boldsymbol{b}_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \left\|\boldsymbol{b}_{i-1}^*\right\|^2 \quad \forall\, 2 \leq i \leq n$$

   puis que

$$\|\boldsymbol{b}_i^*\|^2 \geq 2^{i-j} \left\|\boldsymbol{b}_j^*\right\|^2 \quad \forall\, 1 \leq j < i \leq n\,.$$

2. Montrer que

$$\|\boldsymbol{b}_i\|^2 \leq 2^{i-1} \|\boldsymbol{b}_i^*\|^2 \quad \forall\, 1 \leq i \leq n$$

   puis établir le premier point du théorème.
3. On admet que $\det(\mathcal{L}) = \prod_{i=1}^{n} \|\boldsymbol{b}_i^*\|$. Établir le second point du théorème.
4. On admet que $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ et $(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_k^*)$ engendrent le même espace vectoriel pour tout $k \geq 1$. En écrivant un vecteur $\boldsymbol{v}$ quelconque de $\mathcal{L}$ sur ces bases, établir le troisième point du théorème.

L'Algorithme 3, dit algorithme LLL, permet d'obtenir une base LLL-réduite d'un réseau à partir d'une base quelconque, en un temps polynomial.

Algorithme 3 : Algorithme LLL

Entrée : une base $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ d'un réseau $\mathcal{L} \subseteq \mathbb{R}^n$
Sortie : une base $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ LLL-réduite du réseau $\mathcal{L}$

Pour tout $i = 1, \ldots, n$ **faire**
  $\boldsymbol{b}_i \leftarrow \boldsymbol{a}_i$
$\boldsymbol{b}_1^* \leftarrow \boldsymbol{b}_1$
$k \leftarrow 2$
Tant que $k \leq n$ **faire**
  Pour tout $j = k-1, k-2, \ldots, 1$ **faire**
    Calculer $\mu_{k,j} = \frac{\langle \boldsymbol{b}_k, \boldsymbol{b}_j^* \rangle}{\left\|\boldsymbol{b}_j^*\right\|^2}$
    Remplacer $\boldsymbol{b}_k \leftarrow \boldsymbol{b}_k - \lfloor \mu_{k,j} \rceil \boldsymbol{b}_j$
    Remplacer $\boldsymbol{b}_k^* \leftarrow \boldsymbol{b}_k^* - \mu_{k,j}\, \boldsymbol{b}_j^*$
  Si $\left\|\boldsymbol{b}_k^* + \mu_{k,k-1} \boldsymbol{b}_{k-1}^*\right\|^2 \geq \frac{3}{4} \left\|\boldsymbol{b}_{k-1}^*\right\|^2$
    Incrémenter $k \leftarrow k+1$
  Sinon
    Échanger $\boldsymbol{b}_{k-1}$ et $\boldsymbol{b}_k$
    Définir $k \leftarrow \max\{k-1, 2\}$
Retourner $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$

Question 10.– Implanter l'algorithme LLL. Puis, le tester sur les bases suivantes :

1. La base formée des colonnes de la matrice

$$\boldsymbol{M}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 2 & -1 \\ 1 & 1 & 3 & -1 \\ 0 & 1 & 4 & 1 \end{pmatrix}$$

2. La base formée des colonnes de la matrice

$$\boldsymbol{M}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & -1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & 4 & 1 & 0 \end{pmatrix}$$

(on échangé l'ordre des colonnes)

3. La base formée des colonnes de la matrice

$$\boldsymbol{M}_3 = \begin{pmatrix} 19 & 15 & 43 & 20 & 0 & 48 \\ 2 & 42 & 15 & 44 & 48 & 33 \\ 32 & 11 & 0 & 44 & 35 & 32 \\ 46 & 0 & 24 & 0 & 16 & 9 \\ 3 & 3 & 4 & 18 & 31 & 1 \\ 33 & 24 & 16 & 15 & 31 & 29 \end{pmatrix}$$

Terminons ce sujet en énonçant (sans le démontrer) un résultat de terminaison et de complexité de l'algorithme LLL.

Theorème 3.3

*Soit $\mathcal{L}$ un réseau de dimension n contenu dans $\mathbb{Z}^n$ et $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ une base de $\mathcal{L}$. Alors, l'algorithme LLL termine et produit une base LLL-réduite de $\mathcal{L}$. Par ailleurs, si $B = \max\{\|\boldsymbol{b}_1\|, \ldots, \|\boldsymbol{b}_n\|\}$, alors le nombre de tours de boucles « **Tant que** » de l'algorithme est en $O(n^2 \log n + n^2 \log B)$.*

Références

[BCG+17] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. *Algorithmes Efficaces en Calcul Formel*. Frédéric Chyzak (auto-édit.), 2017.

[LLL82] H.W. Lenstra, A.K. Lenstra, and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261 :515–534, 1982.

[vzGG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra (3. ed.)*. Cambridge University Press, 2013.