TD 12 - Algèbre linéaire

Ce TD n'est pas à préparer à l'avance. Vous le traiterez en binôme pendant la séance.

Nous avons beaucoup travaillé sur des espaces vectoriels sur $\mathbb{R}$ et sur $\mathbb{C}$, mais on peut aussi considérer des espaces vectoriels sur des corps finis. Des exemples simples de corps finis sont les groupes $\mathbb{Z}/p\mathbb{Z}$, où p est un nombre premier.

Exercice 53 : Nombre de bases dans $(\mathbb{Z}/p\mathbb{Z})^n$

Soit p un nombre premier et $n \in \mathbb{N}^*$. Considérons l'espace vectoriel $(\mathbb{Z}/p\mathbb{Z})^n$, muni de l'addition et de la multiplication modulo p.

1. Quel est le cardinal de $(\mathbb{Z}/p\mathbb{Z})^n$? Quel est le cardinal du sous-espace $\mathrm{Vect}(v)$ engendré par un vecteur $v \in (\mathbb{Z}/p\mathbb{Z})^n$? Et celui de $\mathrm{Vect}(v, w)$, où $w \in (\mathbb{Z}/p\mathbb{Z})^n$ est linéairement indépendant de v ?
2. Combien de bases possède l'espace vectoriel $(\mathbb{Z}/p\mathbb{Z})^n$?
   Indication : On peut construire une base vecteur par vecteur.

1. Un élément de $(\mathbb{Z}/p\mathbb{Z})^n$ est un n-uplet de la forme $(a_1, a_2, \ldots, a_n)$ avec $a_i \in \mathbb{Z}/p\mathbb{Z}$.

Il y a p valeurs possibles pour chaque a_i donc $card\left((\mathbb{Z}/p\mathbb{Z})^n\right) = p^n$

$Vect(v) = \{\lambda v,\ \lambda \in \mathbb{Z}/p\mathbb{Z}\}$ ainsi $card(Vect(v)) = p$
$Vect(v, w) = \{\lambda v + \mu w,\ (\lambda, \mu) \in (\mathbb{Z}/p\mathbb{Z})^2\}$ ainsi $card(Vect(v, w)) = p^2$

2. Une base de $(\mathbb{Z}/p\mathbb{Z})^n$ est $\{(1, 0, \ldots, 0) ; (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)\}$

Prenons un vecteur v_1 de $(\mathbb{Z}/p\mathbb{Z})^n$ non nul, il y a $p^n - 1$ choix possibles

Prenons un vecteur v_2 de $(\mathbb{Z}/p\mathbb{Z})^n$ tel que $v_2 \notin Vect(v_1)$, il y a $p^n - p$ choix

Ainsi la famille (v_1, v_2) est libre.

Prenons un vecteur v_3 tel que (v_1, v_2, v_3) soit libre alors $v_3 \notin Vect(v_1, v_2)$, il y a $p^n - p^2$ choix.

Ainsi si $(v_1, \ldots, v_k)$ est libre, $card(Vect(v_1, \ldots, v_k)) = p^k$

Il reste donc $p^n - p^k$ vecteurs qui n'appartiennent pas à $Vect(v_1, \ldots, v_k)$

donc il y a $p^n - p^k$ façons de choisir un vecteur supplémentaire pour notre future base.

En itérant :

$$\#\{bases\ de\ (\mathbb{Z}/p\mathbb{Z})^n\} = \prod_{k=0}^{n-1}(p^n - p^k) = p^{\frac{n(n-1)}{2}} \prod_{k=0}^{n-1}(p^{n-k} - 1) = p^{\frac{n(n-1)}{2}} \prod_{i=1}^{n}(p^i - 1)$$

---

https://www.geogebra.org/classic/de2mgr8n

Exercice 54 : Codes correcteurs

Lors de la transmission d'un message numérique, des erreurs peuvent apparaître. Un code correcteur a pour objectif de coder les messages de façon à permettre au récepteur de savoir si une erreur s'est introduite, et même de la corriger.
Dans la suite, un message consistera en une suite de bits, donc une suite de 0 et de 1, par exemple $m = 0110100010$.

1. Un exemple simple consiste à ajouter un bit à la fin. Quel type de vérification est-ce qu'on peut effectuer dans ce cas ? Donner un exemple.

1. Un exemple simple consiste à ajouter un bit à la fin. Quel type de vérification est-ce qu'on peut effectuer dans ce cas ? Donner un exemple.

$m = 01101\ 00010$
$\alpha(m) = 01101\ 00010$ 0 s'il il y a un nombre pair de 1 dans m
$\alpha(m) = 01101\ 00010$ 1 sinon

2. Le code de Hamming est une méthode plus complexe pour détecter et, éventuellement, corriger des erreurs de transmission. Nous allons en étudier un exemple, le code de Hamming (7,4,3). Nous considérons un message de longueur 4, donc un élément de l'espace vectoriel $(\mathbb{Z}/2\mathbb{Z})^4$. "Coder ce message" signifie alors lui ajouter un certain nombre de bits de contrôle.
Dans cette méthode, le codage et le contrôle sont effectués par des applications linéaires : une application α ajoute des bits au message et une autre application β renvoie 0 si et seulement si le code est correct, dans un sens à préciser.
Considérons le schéma suivant :

$$(\mathbb{Z}/2\mathbb{Z})^4 \xrightarrow[\text{codage}]{\alpha} (\mathbb{Z}/2\mathbb{Z})^k \xrightarrow{\text{transmission}} (\mathbb{Z}/2\mathbb{Z})^k \xrightarrow[\text{contrôle}]{\beta} (\mathbb{Z}/2\mathbb{Z})^l.$$

message — message codé — message transmis — erreur détectée

Les applications α et β doivent vérifier certaines propriétés :

- Deux messages différents ne doivent pas être codés de la même façon par α,
- L'image par β d'un message m_2 transmis est nulle si et seulement s'il s'agit d'un message codé possible, i.e. s'il existe $m \in (\mathbb{Z}/2\mathbb{Z})^4$ tel que $m_2 = \alpha(m)$.

(a) Traduire les propriétés de α et β dans le langage des applications linéaires, notamment en termes de noyau et d'image.

- Comme deux messages différents ne doivent pas être codés de la même façon par α alors α est injective. Ainsi $Ker\ \alpha = \{0\}$.
- L'image par β d'un message m_2 transmis est nulle si et seulement s'il s'agit d'un message codé possible à savoir s'il existe $m \in (\mathbb{Z}/pZ)^4$ tel que $m_2 = \alpha(m)$
  se traduit par : $m_2 \in Ker\ \beta\ ssi\ \beta(m_2) = 0$ ssi $\exists m \in (Z/2Z)^4 / \alpha(m) = m_2$ ssi $m_2 \in Im(\alpha)$
  donc $Ker\ \beta = Im\ \alpha$ et $dim\ Ker\ \beta = rg\ \alpha$

(b) Nous voulons qu'en plus de détecter les erreurs, l'application β nous permette de les corriger. Pour cela, si l'image d'un message par β n'est pas nul, elle doit indiquer quel bit on doit corriger.

i. Supposons que β est surjective. Appliquer le théorème de rang aux applications α et β.

$4 = rg\,\alpha + dim\,Ker\,\alpha = rg\,\alpha$ car $Ker\,\alpha = \{0\}$ donc $dim\,Ker\,\alpha = 0$.

$k = rg\,\beta + dim\,Ker\,\beta = l + dim\,Ker\,\beta$, comme β est surjective alors $rg\beta = dim(\mathbb{Z}/p\mathbb{Z})^l = l$

Avec $dim\,Ker\,\beta = rg\,\alpha = 4$

donc $k = l + 4$

ii. Quelles sont les dimensions minimales possibles k et l des espaces vectoriels intervenant dans le schéma ?

Indication : Commencer par déterminer l.

Dans l'application β :

$$(\mathbb{Z}/2\mathbb{Z})^k \xrightarrow[\text{contrôle}]{\beta} (\mathbb{Z}/2\mathbb{Z})^l.$$

message transmis — erreur détectée

A un k-uplet, on associe la place de l'erreur, à savoir un nombre compris entre 1 et k.

Or $card((\mathbb{Z}/2\mathbb{Z})^l) = 2^l$ donc on doit avoir que $2^l \geq k$

Nous savons que : $l = k - 4$,

on cherche k tel que $2^{k-4} \geq k$ à savoir $2^k \geq 16k$

pour $k = 6 : 2^6 = 64 < 16 \times 6 = 96$

pour $k = 7 : 2 = 128 > 16 \times 7 = 112$

donc k est supérieur ou égal à 7.

On en conclut que comme $l = k - 4$, alors l est supérieur ou égal à 3

(c) Soit maintenant $m \in (\mathbb{Z}/2\mathbb{Z})^4$ un message et soient $m_1 = \alpha(m) \in (\mathbb{Z}/2\mathbb{Z})^k$ le message codé et m_2 le message transmis. S'il y a une erreur dans la transmission du i-ième bit, cela signifie que le récepteur a reçu : $m_2 = m_1 + e_i$, où e_i est le i-ième vecteur de la base canonique de $(\mathbb{Z}/2\mathbb{Z})^k$, p.ex. $e_1 = (1, 0, \ldots, 0)$.
Justifier que $\beta(m_2) = \beta(e_i)$.

$\beta(m_2) = \beta(m_1 + e_i) = \beta(m_1) + \beta(e_i)$ par linéarité de β

or $\beta(m_1) = 0$ car $m_1 = \alpha(m)$ et $Ker\,\beta = Im\,\alpha$

donc $\beta(m_2) = \beta(e_i)$

(d) Supposons $k = 7, l = 3$.

i. Exprimer les entiers de 1 à 7 en base 2.

1 : 001
2 : 010
3 : 011
4 : 100
5 : 101
6 : 110
7 : 111

ii. Construire une matrice B pour l'application β qui convienne.
Rappel : Idéalement, on aimerait qu'en cas d'erreur, β nous indique quel bit il faut corriger.

On doit avoir que $\beta(e_i) = ième\ place\ de\ l'erreur$

ainsi $\beta(e_1) = 1$ (1 écrit en décimal) mais si on l'écrit en binaire $\beta(e_1) = 001 \in (\mathbb{Z}/2\mathbb{Z})^3$

$\beta(e_2) = 010$, …, en utilisant l'écriture binaire des nombres de 1 à 7.

Ce qui donne la matrice B :

$$B = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

(e) Construire ensuite une matrice A pour l'application α qui convienne.
Rappel : A doit être de rang 4 et vérifier BA = 0.

- Pour la matrice A, on sait que les quatre premières lignes donnent le message inchangé, donc c'est la matrice identité, elle a trois lignes supplémentaires qui correspondent au contrôle.

$$\underset{\text{message}}{(\mathbb{Z}/2\mathbb{Z})^4} \xrightarrow[\text{codage}]{\alpha} \underset{\text{message codé}}{(\mathbb{Z}/2\mathbb{Z})^k}$$

Par exemple : $(1\ 0\ 0\ 0) \rightarrow (1\ 0\ 0\ 0\ 1\ 1\ 1)$ avec ici $k = 7$
donc sa matrice est de la forme :

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_2 \end{pmatrix}$$

- La matrice A doit aussi respecter $BA = 0$ car $\beta \circ \alpha = 0$ (du fait que $Ker\,\beta = Im\,\alpha$)

$$BA = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{pmatrix} =$$

$$\begin{pmatrix} a_1+a_2+a_3 & b_1+b_2+b_3 & c_1+c_2+c_3 & 1+d_1+d_2+d_3 \\ a_2+a_3 & 1+b_2+b_3 & 1+c_2+c_3 & d_2+d_3 \\ 1+a_1+a_3 & b_1+b_3 & 1+c_1+c_3 & d_1+d_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

On obtient les systèmes suivants à résoudre

$$\begin{cases} a_1+a_2+a_3=0 \\ a_2+a_3=0 \\ 1+a_1+a_3=0 \end{cases} \Leftrightarrow \begin{cases} a_1=0 \\ a_2=1 \\ a_3=1 \end{cases} \quad \begin{cases} c_1+c_2+c_3=0 \\ 1+c_2+c_3=0 \\ 1+c_1+c_3=0 \end{cases} \Leftrightarrow \begin{cases} c_1=1 \\ c_2=1 \\ c_3=0 \end{cases}$$

$$\begin{cases} b_1+b_2+b_3=0 \\ 1+b_2+b_3=0 \\ b_1+b_3=0 \end{cases} \Leftrightarrow \begin{cases} b_1=1 \\ b_2=0 \\ b_3=1 \end{cases} \quad \begin{cases} d_1+d_3=0 \\ 1+d_1+d_2+d_3=0 \\ d_2+d_3=0 \end{cases} \Leftrightarrow \begin{cases} d_1=1 \\ d_2=1 \\ d_3=1 \end{cases}$$

Ainsi la matrice A est :

$$A=\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&1&1&1 \\ 1&0&1&1 \\ 1&1&0&1 \end{pmatrix}$$

(f) Choisir un message $m \in (\mathbb{Z}/2\mathbb{Z})^4$ et tester la chaîne codage/transmission/contrôle, en perturbant le message codé lors de la transmission. Est-ce que l'erreur est détectée et corrigée ? Est-ce que cela fonctionne quelle que soit la perturbation choisie ? Que se passe-t-il s'il y a 2 erreurs ? Plus que 2 erreurs ?

Exemple avec 1 erreur :

On choisit le message $m = (1\ 0\ 0\ 0)$
On calcule $m_1 = \alpha(m)$, soit le calcul :

$$AM=\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&1&1&1 \\ 1&0&1&1 \\ 1&1&0&1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}=\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$ ainsi $m_1 = \alpha(m) = (1\ 0\ 0\ 0\ 0\ 1\ 1)$

On suppose que le message transmis est : $m_2 = (1\ 0\ 0\ 1\ 0\ 1\ 1)$ avec l'erreur sur la 4e place.
On calcule $\beta(m_2)$, soit le calcul :

$$BM_2=\begin{pmatrix} 0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}=\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

or 1 0 0 correspond au nombre 4, l'erreur est donc bien en 4e position.

Exemple avec 2 erreurs :

On choisit le message $m = (1\ 0\ 0\ 0)$
On calcule $m_1 = \alpha(m)$, soit le calcul :

$$AM = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$ ainsi $m_1 = \alpha(m) = (1\ 0\ 0\ 0\ 0\ 1\ 1)$

On suppose que le message transmis est : $m_2 = (0\ 0\ 0\ 1\ 0\ 1\ 1)$ avec l'erreur en positions 1 et 4.

On calcule $\beta(m_2)$, soit le calcul :

$$BM_2' = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Or (1 0 1) correspond à 5 en binaire, on a additionné les rangs des erreurs.
Ce code contrôle n'est pas valable pour deux erreurs.