M1
ALGÈBRE LINÉAIRE POUR L'AGRÉGATION (TRON COMMUN)

CHRISTOPHE RITZENTHALER

Petite intro du cours de Schost [1, Chap.8, sec.1]. Puis on commence par l'algèbre linéaire sur un corps K avec des rappels sur les méthodes basiques. Ensuite on passe à la multiplication rapide et à ses conséquences. Enfin, on étudie les problématiques sur un anneau euclidien.

1. Rappels

Les résultats suivants ont été vus dans le cours "systèmes linéaires". Nous les incluons pour mémoire.

Dans cette partie, A est une matrice carrée d'ordre n inversible à coefficients dans un corps K.

1.1. **Règles de Cramer.** Afin de résoudre $AX = Y$ lorsque A est inversible, on peut utiliser les règles de Cramer. En effet puisque

$$X = A^{-1}Y = \frac{1}{\det(A)}{}^t A^{co} Y = \frac{1}{\det(A)} \begin{pmatrix} \cdots \\ \sum y_i cof_{ij} \\ \cdots \end{pmatrix}$$

La somme entre parenthèse est donc le développement en cofacteurs de $\det(A_i)$ par rapport à sa i-ème colonne où A_i est la matice où on a remplace la i-ème colonne de A par le vecteur colonne Y.
La complexité de la méthode est celle du calcul de $n+1$ déterminants. Elle est donc à proscrire en général (mais est utile pour voir comment varie une solution par exemple ou dans des études de complexité).

1.2. **Pivot de Gauss.** On utilise l'algorithme ci-dessous, qui permet de résoudre des systèmes linéaires, calculer un rang, un déterminant ou un inverse (voir à la fin) en un coût $O(n^3)$.

Entree : $A \in M_n(k)$ inversible.
Sortie : A triangulaire. **for** *i=1..n* **do**
  Chercher un pivot dans la colonne C_i :
  chercher $j \geq i$ tq $a_{ji} \neq 0$
  Echanger L_i et L_j
  On annule les coefs sous a_{ii} :
  for *j=i+1..n* **do**
    $L_j \leftarrow L_j - a_{ij}/a_{ii} L_i$
Renvoyer A

Calcul du coût : $O(n^3)$. On peut tenir compte d'un vecteur Y pour résoudre le système $AX = Y$.

L'interprétation matricielle de l'algorithme précédent : on a transforme le système $AX = Y$ en $PAX = Pb$ en faisant des opérations sur les lignes, et la matrice $A' = PA$ est triangulaire supérieure. On la note U. S'il se trouve qu'a chaque fois on choisit le premier pivot (i.e. on ne fait aucun échange de lignes), alors P est triangulaire inférieure, avec des 1 diagonaux. On la note L. Son coef L_{ji} est le coefficient a_{ij}/a_{ii} utilise dans l'élimination.

Théorème 1.1. *Une matrice inversible $A \in GL_n(k)$ admet une décomposition $A = LU$ avec U triangulaire supérieure, L triangulaire inférieure avec diagonale 1 ssi chaque sous-matrice $\Delta_k = A_{1\leq i,j\leq k}$ est inversible. Lorsque cette decomposition existe elle est unique, et l'algorithme modifie ci-dessus la calcule.*

Entree : $A \in M_n(k)$.
Sortie : decomposition LU
Initialiser $L = \mathrm{Id}$. **for** *i=1..n* **do**
  Si $a_{ii} = 0$, renvoyer "Erreur, LU n'existe pas"
  On annule les coefficients sous a_{ii} :
  for *j=i+1..n* **do**
    $L_{ji} \leftarrow a_{ij}/a_{ii}$
    effectuer sur A $L_j \leftarrow L_j - a_{ij}/a_{ii}L_i$
$U \leftarrow A$ Renvoyer L, U

Un cas particulier important est celui où A est symétrique définie positive (tous les sous-mineurs diagonaux sont symétriques définies positifs – en regardant cela sur des vecteurs avec des 0 en queue–).

Remarque 1.2. Quand A ne remplit pas nécessaire la condition du théorème 1.1, il existe une décomposition LUP où P est une matrice de permutation qui tient compte de la recherche d'un pivot dans la *ligne*.

Entrée : $A \in M_n(k)$ inversible
Sortie : LUP
Initialiser $L, P \leftarrow \mathrm{id}$
$U \leftarrow A$
for *i=1..n* **do**
  Chercher un pivot dans la ligne L_i
  chercher $j \geq i$ tel que $U_{ij} \neq 0$
  Echanger C_i et C_j
  $P \leftarrow \sigma P$
  On annule les coefficient sous a_{ii} :
  for *j=i+1..n* **do**
    $L_{ji} \leftarrow U_{ij}/U_{ii}$
    $L_j \leftarrow L_j - U_{ij}/U_{ii}L_i$
Renvoyer L, U, P

Pour le calcul de l'inverse de A, on utilise une variante de l'algorithme de Gauss, dite de Gauss-Jordan qui produit une forme échelonnée réduite de A : il s'agit d'une forme

échelonnée de A dont les pivots valent 1, les autres coefficients dans les colonnes des pivots étant nuls. Appliquée à la matrice $\tilde{A} = [A|\mathrm{Id}]$, cette variante transforme la matrice $\tilde{A}$ en une matrice équivalente dont le bloc de gauche est l'identité, c'est-à-dire qu'elle remplace $\tilde{A}$ par $[\mathrm{Id}|A^{-1}]$.

Remarque 1.3. Le même algorithme permet de résoudre $AX = Y$ en bordant la matrice A par le vecteur b.

1.3. **Choix du pivot, conditionnement.** Ici on se place sur $\mathbb{R}$ (le cas complexe est analogue). On veut garer les erreurs d'arrondi. On se donne $||.||$ une norme sur $\mathbb{R}^n$, $||.||$ la norme matricielle subordonnée associée.

Définition 1.4. Soit A une matrice inversible, et $||.||$ une norme subordonnée sur $M_n(\mathbb{R})$. $\mathrm{cond}(A) = ||A||.||A^{-1}||$

Remarque 1.5. C'est toujours un nombre ≥ 1.

Théorème 1.6. *Soit A inversible, $b \in \mathbb{R}^n \setminus 0$, u solution de $Au = b$. Soit $\delta b \in \mathbb{R}^n$, et $u + \delta u$ la solution de $A(u + \delta u) = b + \delta b$. Alors $||\delta u||/||u|| \leq cond(A)||\delta b||/||b||$. De plus, il existe $b, \delta b$ avec égalité.*

Démonstration. $A(u + \delta u) = b + \delta b$ donc $A\delta u = \delta b$ et $\delta u = A^{-1}\delta b$. On veut $||\delta u||.||b|| \leq (||a^{-1}||||\delta b||)(||A||||u||)$. OK car $\delta u = A^{-1}\delta b$ et $b = Au$. □

Théorème 1.7. *Soit A inversible, $b \in \mathbb{R}^n \setminus 0$, u solution de $Au = b$. Soit $(A + \delta A)(u + \delta u) = b$ pour une certaine matrice δA, et δu. Alors $\frac{||\delta u||}{||u+\delta u||} \leq cond(A)||\delta A||/||A||$. De plus, il existe $b, \delta A$ avec égalité.*

Exemple 1.8. Exemples de conditionnement :
- Pour $||.||_2$: Si A isometrie, $\mathrm{cond}(A) = 1$.
- Si A est symétrique $\mathrm{cond}(A) = \max|vap|/\min|vap|$.
- Si $||.||_\infty$, alors $||A|| = sup_i \sum_j |a_{ij}| = \max_i ||L_i||_1$.

Un pas d'élimination consiste a changer A en EA, avec $\lambda_j = -a_j/a_i$. Comme on a que $\mathrm{cond}(EA) \leq \mathrm{cond}(E)\mathrm{cond}(A)$, donc on veut $\mathrm{cond}(E)$ minimal, i.e. $|a_j/a_i|$ minimal, d'où le choix de pivot de valeur absolue maximale.

2. Plus vite que $O(n^3)$

2.1. **Algorithmes de Winograd et Strassen.** Voir [1, Chap.8, sec.2].

2.2. **Inverse, déterminant et polynôme caractéristique.** Voir [1, Chap.8, sec.3].

2.3. **Algèbre linéaire creuse.** Voir [1, Chap.9].

3. Sur un anneau euclidien

3.1. **Cadre.** On se place sur $\mathbb{Z}$ ou sur $k[X]$ avec k corps. Plus généralement, sur un anneau euclidien R.

Rappel : R est euclidien s'il est intègre et s'il existe une jauge euclidienne $\delta : R\setminus\{0\} \to \mathbb{N}$, $\delta(0) = -\infty$

(1) $\forall a, b, b \neq 0\ \exists q, r$ tq $a = bq + r,\ \delta(r) < \delta(b)$

(2) $a|b, b \neq 0 \Rightarrow \delta(a) \leq \delta(b)$. [1]

1. La seconde condition n'est pas nécessaire pour définir un anneau euclidien : s'il existe une application vérifiant (1) alors on peut définir une application vérifiant (1) et (2).

Ex : $k[X]$, degré, $\mathbb{Z}, |.|$ (modifiée en 0).

Exemple 3.1. si $b|a$, et si $a = bq + r$ est une division euclidienne, alors $r = 0$. Indication : b divise le reste, donc $\delta(b) \leq \delta(r)$ si $r \neq 0$.

On suppose qu'on a un algorithme pour la division euclidienne, donc pour la division tout court si $a|b$.

Rappel : euclidien implique intègre, et principal (prendre un élément de jauge minimale). En particulier, on a Bézout $\forall a, b \exists u, v | au + bv = a \wedge b$, et l'algorithme d'Euclide étendu permet de calculer $u, v, a \wedge b$.

Définition 3.2. $A \sim_d B$ si $AP = B$ avec $P \in GL_n(R)$, $A \sim_g B$ si $PA = B$ avec $P \in GL_n(R)$.

Remarque 3.3. $A \in GL_n(R)$ ssi $\det A$ inversible dans R donc dans $GL_n(\mathbb{Z})$ ssi $\det = \pm 1$, dans $GL_n(k[X])$ ssi $\det \in k^*$.

3.2. Échelonnement, forme de Hermite.

Théorème 3.4. *Soit $A \in M_{np}(R)$, R euclidien. Alors $A \sim_g Ech$ avec Ech échelonnée en lignes, et il y a un algorithme de calcul.*

3.2.1. *Version avec divisions euclidiennes.* On a l'algorithme suivant.

Entrée : $A_0 \in M_{np}(R)$.
Sortie : $A \sim_g A_0$, échelonnée
$A \leftarrow A_0$, $i_0 \leftarrow 1, j_0 \leftarrow 1$,
while $i_0 \leq n$ *et* $j_0 \leq p$ **do**
  Rechercher pivot $a_{ij_0} \neq 0$, $i \geq i_0$ dans la colonne C_{j_0}
  Si y en a pas : $j_0 \leftarrow j_0 + 1$, recommencer le tant que.
  Sinon, $L_i \leftrightarrow L_{i_0}$.
  Élimination :
  $i \leftarrow i_0 + 1$ **while** $i \leq n$ **do**
    Faire $L_i \leftarrow L_i - qL_{i_0}$ où $a_{ij_0} = qa_{i_0j_0} + r$, $\delta(r) < \delta(a_{i_0j_0})$
    [Note : après opération $a_{ij_0} = r$]
    Si $r \neq 0$, échanger L_i et L_{i_0}, [et recommencer la boucle avec le même i]
    Si $r = 0$: $i \leftarrow i + 1$
  $i_0 + +, j_0 + +$
renvoyer A.

Remarque 3.5. Il existe une forme normale qui assure l'unicité également.

3.2.2. *Lien avec l'algorithme d'Euclide étendu.* Si on applique cet algorithme avec une matrice colonne $A = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$, on obtient en sortie une matrice $A' = \begin{bmatrix} d \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, et une matrice P tq $PA = \begin{bmatrix} d \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

Lemme 3.6. *d est le pgcd de $a_1, \ldots, a_n$. La première ligne de P donne des coefficients de Bézout pour les a_i : $\sum P_{1i} a_i = d$.*

Démonstration. Puisque $A = P^{-1}A'$, on a que d divise tous les coefficients de a donc $d|\operatorname{pgcd}(a_1, \ldots, a_n)$. Réciproquement, si k divise tous les a_i, alors k divise $PA = A'$, donc k divise d. □

En fait, dans le cas où $n = 2$, on retrouve exactement l'algorithme d'Euclide étendu qui permet de calculer les coefficients de Bézout :

Corollaire 3.7. *Étant donnés a, b, on peut determiner le pgcd $d = a \wedge b$, et une matrice 2×2 $P = \begin{bmatrix} u & v \\ u' & v' \end{bmatrix}$ inversible, telle que $P.\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix}$, i.e $au + bv = d$, et $au' + bv' = 0$.*

Remarque 3.8. Les coefficients u', v' ne sont pas mystérieux : $u' = -b/d, v' = a/d$ conviennent.

3.2.3. *Version avec les coefficients de Bézout.* Au lieu de faire une suite d'opérations de division euclidiennes, on va utiliser les matrices 2×2 P ci-dessus.

Définition 3.9. Soit $\begin{bmatrix} u & v \\ u' & v' \end{bmatrix} \in GL_2(R)$ une matrice inversible. On appelle matrice pseudo-élémentaire $E_{i_1,i_2}(u, v, u', v')$ la matrice (e_{ij}) tq $e_{i,j} = \delta_{i,j}$ si $i \notin \{i_1, i_2\}$ ou $j \notin \{i_1, i_2\}$ et $e_{i_1,i_1} = u$, $e_{i_1,i_2} = v$, $e_{i_2,i_1} = u'$, $e_{i_2,i_2} = v'$.

On appelle opération pseudo-élémentaire sur les lignes d'une matrice A la multiplication a gauche par une matrice pseudo-élémentaire.

Informellement, on note $\begin{bmatrix} L_{i_1} \\ L_{i_2} \end{bmatrix} \leftarrow \begin{bmatrix} u & v \\ u' & v' \end{bmatrix} \begin{bmatrix} L_{i_1} \\ L_{i_2} \end{bmatrix}$.

Ca donne une nouvelle version de l'algorithme dans lequel on ne fait qu'un calcul de Bézout.

Entree : $A_0 \in M_{np}(R)$.
Sortie : $A \sim_g A_0$, échelonnée
$A \leftarrow A_0$, $i_0 \leftarrow 1, j_0 \leftarrow 1$,
while $i_0 \leq n$ *et* $j_0 \leq p$ **do**
  Rechercher pivot $a_{ij_0} \neq 0$, $i \geq i_0$ dans la colonne C_{j_0}
  Si y en a pas : $j_0 \leftarrow j_0 + 1$, recommencer le tant que.
  Sinon, $L_i \leftrightarrow L_{i_0}$.
  Élimination :
  for i *de* $i_0 + 1$ *a* n **do**
    Calculer $\begin{bmatrix} u & v \\ u' & v' \end{bmatrix} \in GL_2(R)$ tq pour $\begin{bmatrix} u & v \\ u' & v' \end{bmatrix} . \begin{bmatrix} a_{i_0 j_0} \\ a_{ij_0} \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix}$. Faire
    $\begin{bmatrix} L_{i_0} \\ L_i \end{bmatrix} \leftarrow \begin{bmatrix} u & v \\ u' & v' \end{bmatrix} . \begin{bmatrix} L_{i_0} \\ L_i \end{bmatrix}$
  $i_0 + +, j_0 + +$
renvoyer A.

3.3. **Applications de la forme de Hermite.**

3.3.1. *PGCD.* La forme de Hermite généralisée le pgcd : $\begin{bmatrix} a_1 \\ \ldots \\ a_n \end{bmatrix} \sim_g \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix}$. Et en fait $d = pgcd(a_1, \ldots, a_n)$.

3.3.2. *Base de l'image.* Rappel : une base d'un sous-module $M \subset R^n$ est une famille libre, génératrice : l'application de $F : (\lambda_i) \in R^k \to \sum_i \lambda_i \vec{v_i}$ est injective et surjective.

Soit $A \in M_{np}(R)$, $A \sim_d Ech$, avec Ech échelonnée en colonnes. $\operatorname{Im} A = \operatorname{Im} Ech$ et les colonnes non nulles de Ech sont linéairement indépendantes.

Corollaire 3.10. *Tout sous-module de type fini de R^n admet une base. L'algorithme de Hermite permet de trouver une base à partir d'un système fini de générateurs.*

Démonstration. Dire qu'un module est de type fini, c'est dire que c'est l'image d'une matrice A. Puis échelonner en colonnes. □

Remarque 3.11. En fait, on peut montrer que tout sous-module est de type fini (car R principal donc est noethérien).

Remarque 3.12. Le corolllaire est faux pour $R = k[x, y]$. Prendre le module $M =< x, y >$ correspondant à la matrice $\begin{bmatrix} x & y \end{bmatrix}$. Cette matrice n'est pas équivalente à une matrice échelonnée en colonnes.

3.3.3. *Base du noyau.* On échelonne en colonnes : $AP = Ech$. $AX = 0 \iff EchP^{-1}X = 0 \iff P^{-1}X \in \ker Ech \iff X \in P(\ker Ech)$, donc $\ker A = P \ker Ech$. Donc suffit de trouver une base de $kerEch$ pour Ech ech en colonnes. Mais les colonnes non nulles sont linéairement indépendantes, donc $kerEch$ a pour base $e_{r+1}, \ldots, e_n$, donc une base de $kerA$ est donnée par les $n - r$ derniers vecteurs de P.

3.3.4. *Solution de $AX = b$ dans $\mathbb{Z}$.* On a le noyau, donc il faut trouver une solution particulière, ou vérifier qu'il n'y en a pas. On pose $X = PY$ avec $AP = Ech$ en colonnes. $AX = b \iff APY = b \iff EchY = b$. Il s'agit donc de trouver une solution particulière dans le cas $EchY = b$.

Méthode a la main : les équations des pivots déterminent les variables par des équations de la forme $a'x' + ax = b$ avec $a'x'$ déjà connu, donc il s'agit de savoir si a divise $b - a'x'$, et si oui de faire la division. Ensuite on a d'autres équations dont on vérifie qu'elles sont satisfaites, etc.

3.4. **Diagonalisation/forme de Smith.**

Théorème 3.13. *Toute matrice $A \in M_{np}(R)$ est équivalente à une matrice par blocs $\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$ avec $D = diag(d_1, \ldots d_r)$ et $d_1 | d_2 \ldots d_r$ non nuls, et il existe un algorithme.*

Unicité : si A équivalente à un autre telle forme, $r = r'$, et $d_i \sim d'_i$ (associés). On appelle les d_i les facteurs invariants *de A.*

3.5. **Applications de Smith.**

3.5.1. *Décider si 2 matrices sont équivalentes.* Calculer la forme de Smith et comparer les d_i, en utilisant l'unicité.

3.5.2. *Théorème de la base adaptée.*

Théorème 3.14. *Soit $M \subset R^n$ un sous-module de type fini. Il existe une base $v_1, \ldots v_n$ de R^n, et $d_1 | d_2 .. | d_r$ tq $M =< d_i v_i >$.*

Si on se donne des générateurs de M, on peut trouver v_i, d_i.

Démonstration. Ecrire $R = \operatorname{Im} A$ et appliquer Smith à A : $A = PDQ$. $\operatorname{Im} A = \operatorname{Im} PD$, et on peut prendre $v_i = Pe_i =$ les vecteurs colonnes de P : $\operatorname{Im} A =< d_1 v_1, ..., d_r v_r >$. □

0. Si $A = 0$, rien a faire. Sinon, échanger lignes et colonnes, pour que $a_{11} \neq 0$, et $\delta(a_{11})$ minimal parmi les $\delta(a_{ij})$.
1. Pour chaque i de 2 a n,
– si $a_{11} \not| a_{i1}$:
  (1) faire la division euclidienne $a_{i1} = a_{11}q + r$, avec $\delta(r) < \delta(a_{11})$
  (2) faire $L_i \leftarrow L_i - qL_1$ [maintenant $a_{i1} = r$, donc $\delta(a_{i1}) < \delta(a_{11})$]
  (3) Retourner en 0.
– si $a_{11}|a_{i1}$,
  (1) faire $L_i \leftarrow L_i - qL_1$ [fait apparaitre un 0 en a_{i1}] ;
  (2) passer au i suivant.
2. Même chose pour les colonnes : pour chaque j de 2 a n,
– si $a_{11} \not| a_{1j}$:
  (1) faire la division euclidienne $a_{1j} = a_{11}q + r$, avec $\delta(r) < \delta(a_{11})$
  (2) faire $C_j \leftarrow C_j - qC_1$ [maintenant $a_{1j} = r$, donc $\delta(a_{1j}) < \delta(a_{11})$]
  (3) Retourner en 0.
– si $a_{11}|a_{1j}$,
  (1) faire $C_j \leftarrow C_j - qC_1$ [fait apparaitre un 0 en a_{1j}] ;
  (2) passer au j suivant.
3. Si a_{11} ne divise pas un certain coef $a = a_{ij}$:
  (1) faire $C_1 \leftarrow C_1 + C_i$, [pour faire apparaitre a en a_{i1}]
  (2) div euclidienne $a = a_{11}q + r$, $\delta(r) < \delta(a_{11})$,
  (3) faire $L_i \leftarrow L_i - qL_1$, [maintenant $a_{i1} = r$, donc $\delta(a_{i1}) < \delta(a_{11})$
  (4) retourner en 0.
4. Recommencer avec la sous-matrices en bas a droite.

Corollaire 3.15. *Tout groupe abélien de type fini s'écrit sous la forme* $Z/d_1Z \oplus \cdots \oplus \mathbb{Z}/d_r\mathbb{Z} \oplus Z^k$. *Avec* $d_i \geq 2$, *et* $d_1|d_2|\ldots|d_r$.

Si on écrit $G = \mathbb{Z}^n/ < v_1, \ldots, v_i >= Z^n/Im\, A$, *on peut algorithmiquement calculer les* n_i *et* r.

3.5.3. *Matrices semblables.* Ici on se place sur $k[X]$, k corps ($k = \mathbb{Q}$).
Rappel : lien $k[X]$-module et k-ev muni d'un endomorphisme :

(1) Si E est un k-ev muni d'un endomorphisme u, E peut-être muni d'une structure de $k[X]$-module en définissant pour tout polynôme P, $P.\vec{v} = P(u).\vec{v}$. En particulier, $X.\vec{v} = u.\vec{v}$.

(2) Si E est un $k[X]$-module, c'est en particulier un k-ev ; de plus il a un endomorphisme canoniquement associe : c'est l'endomorphisme $u : E \to E$ défini par $u(\vec{v}) = X.\vec{v}$. Ces 2 opérations sont réciproques l'une de l'autre.

(3) $u : E \to E$ et $u' : E' \to E'$ sont conjugues ssi il existe $f : E \to E'$ un isomorphisme de k-ev tq $f(u(\vec{)}) = u'(f(\vec{v}))$ pour tout $\vec{v} \in E$, ssi, $f(P(u).\vec{v}) = P(u').f(\vec{v})$, ie dans le langage des modules : $f(P\vec{v}) = Pf(\vec{v})$ ie f est $k[X]$-linéaire, ie c'est un isomorphisme de $k[X]$ module.

Théorème 3.16. *On a les équivalences*

(1) A semblable a B

(2) $A - XI$ est équivalente a $B - XI$ dans $M_n(k[X])$

(3) $k[X]^n/Im(A - XI) \simeq k[X]^n/Im(B - XI)$ (comme $k[X]$-module)

De plus, les facteurs invariants de la matrice $\tilde{A} = A - XI$ sont (en oubliant les polynômes de degré 0) les invariants de similitude de A.

Corollaire 3.17. *On peut calculer algorithmiquement les facteurs invariants d'une matrice à coefficients dans $\mathbb{Q}$ ou dans un corps fini. On peut donc déterminer si 2 matrices données sont semblables ou non.*

Exemple 3.18. Donner la forme de Jordan de la matrice ci-dessous pui en déduire la liste des invariants de similitude

$$\begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 5 \\ 0 & 0 & 0 & 0 & 0 & 7 \end{bmatrix}.$$

Solution : La matrice a deux valeurs propres distinctes. Soit u l'endomorphisme associé à M. Puisque $\dim \ker(u-1) = 1$ il n'y a qu'un seul bloc de Jordan $J_3(1)$ et puisque $\dim \ker(u-7) = 2$ il y a deux blocs de Jordan associé à cette valeur propre donc puisque la somme de leur dimension est 3, il y a un bloc $J_1(7)$ et un bloc $J_2(7)$.
Les invariants de similitude associés aux blocs de Jordan sont $(X-7), (X-7)^2$ et $(X-1)^3$.
La structure de $K[X]$-module de K^6 est donc

$$\frac{K[X]}{(X-7)} \times \frac{K[X]}{(X-7)^2} \times \frac{K[X]}{(X-1)^3} \simeq \frac{K[X]}{(X-7)} \times \frac{K[X]}{(X-7)^2(X-1)^3}.$$

Donc les invariants de similitude sont $(X-7)$ et $(X-7)^2(X-1)^3$.

3.5.4. *Algorithme pour compléter en une base.* On se place sur $\mathbb{Z}$ pour être plus concret.

Entrée : $v_1, \dots v_r$. La question est de savoir si on peut compléter en une base, et comment.

Proposition 3.19. *$v_1, \dots, v_r$ se complètent en une base ssi les d_i du théorème de la base adaptée applique à $< v_1, .., v_r >$ sont tous égaux a 1, et $r = rang(A)$.*

Références

[1] A. Bostan et al. : *Algorithmes efficaces en calcul formel* disponible sur `https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-22`

[2] H. Cohen : *A course in computational algebraic number theory*, Springer-Verlag, 1993.