Université Bordeaux 1 **PIN 401**
Mathématiques Année 2012–2013

FEUILLE D'EXERCICES n° 2 (complément)

Le but de ce document est de montrer à moindre frais comment on peut formaliser et systématiser les notions et opérations vues en TD sur des exemples. Il n'est pas question ici de minimiser le nombre d'opérations dans le corps K ($= \mathbb{R}$ ou $\mathbb{C}$) ou l'erreur numérique. Si M est une matrice dont la plus grande dimension (lignes ou colonnes) est n, toutes les algorithmes qui suivent utilisent $O(n^3)$ opérations dans K (de type $\pm, \times, /$). On peut faire mieux si n est grand, mais en particulier si la matrice M est très structurée[1].

1. Opérations sur les colonnes (image)

Définition 1. La matrice $m \times n$ (m lignes et n colonnes) $M = (M_{i,j})$ est *échelonnée en colonnes* s'il existe une application strictement croissante $f : [1, n] \to [1, m]$ telle que

- Pour $1 \leqslant j \leqslant n$, on a $M_{f(j),j} = 1 \quad (M_{f(j),j} \neq 0$ nous suffirait),
- Pour $i < f(j)$, on a $M_{i,j} = 0$,
- Pour $k > j$, on a $M_{f(j),k} = 0$.

Les $M_{f(j),j} = 1$, pour $j = 1, \dots, n$ sont les *pivots* de M. En d'autres termes, à droite (même ligne) et au dessus (même colonne) d'un pivot, il n'y a que des 0, et les pivots forment une ligne brisée décroissante ; $f(j)$ donne la ligne du j-ème pivot (celui de la j-ème colonne).

Exemple : ici, $f(1) = 1$, $f(2) = 3$, $f(3) = 5$, $f(4) = 6$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ \times & 0 & 0 & 0 \\ \times & 1 & 0 & 0 \\ \times & \times & 0 & 0 \\ \times & \times & 1 & 0 \\ \times & \times & \times & 1 \\ \times & \times & \times & \times \end{pmatrix}$$

On note $M_{i,\bullet}$ la i-ème ligne et $M_{\bullet,j}$ la j-ème colonne de M.

Théorème 2. *Les colonnes d'une matrice échelonnée en colonnes forment une famille libre.*

Preuve. Si $\lambda_1 M_{\bullet,1} + \dots \lambda_n M_{\bullet,n} = 0$, alors la ligne d'indice $f(1)$ dit $\lambda_1 + 0 + \cdots + 0 = 0$, soit $\lambda_1 = 0$. De proche en proche on prouve que les λ_i sont tous nuls. $\square$

[1] Beaucoup de 0 dans des positions faciles à décrire, par exemple M diagonale !

Algorithme . Image d'une application linéaire φ

Entrées: $M = (M_{i,j}) \in \mathcal{M}_{m\times n}(K)$ la matrice $\mathrm{Mat}(\varphi, \mathcal{B}, \mathcal{B}')$.
Sorties: Une base échelonnée en colonnes de $\mathrm{Im}\,\varphi$ (donnée dans la base $\mathcal{B}'$).

Poser $j \leftarrow 0$.
pour $i = 1, 2, \ldots, m$ **faire** $\{$*on traite la ligne* $i\}$
  si il existe $p > j$, tel que $M_{i,p} \neq 0$ **alors** $\{$*un pivot sur cette ligne ?*$\}$
    Choisir p tel que $M_{i,p} \neq 0$ comme *pivot*. Par exemple, $|M_{i,p}|$ minimal.
    Poser $j \leftarrow j + 1$.
    Échanger les colonnes $M_{\bullet,p}$ et $M_{\bullet,j}$.
    Diviser la colonne $M_{\bullet,j}$ par $M_{i,j}$. $\{$*le pivot* $M_{i,j}$ *est en place,* $f(j) = i\}$
    pour $k = j + 1, \ldots, n$ **faire** $\{$*annuler* $M_{i,k}$ *en utilisant* $M_{i,j} = 1\}$
      $M_{\bullet,k} \leftarrow M_{\bullet,k} - M_{i,k} \times M_{\bullet,j}$
Retourner les j premières colonnes de M. L'image est de dimension j.

Comme $p > j$ lors du choix du pivot, la colonne p ne contient pas déjà un pivot.

2. Opérations sur les lignes (noyau)

Définition 1. La matrice $m \times n$ (m lignes et n colonnes) $M = (M_{i,j})$ est *échelonnée en lignes* si sa transposée $(M_{j,i})$ est échelonnée en colonnes.

Autrement dit, s'il existe une application strictement croissante $g : [1, m] \to [1, n]$ telle que

- Pour $1 \leqslant i \leqslant m$, on a $M_{i,g(i)} = 1$ $\quad (M_{i,g(i)} \neq 0$ nous suffirait),
- Pour $j < g(i)$, on a $M_{i,j} = 0$,
- Pour $k > i$, on a $M_{k,g(i)} = 0$.

Les $M_{i,g(i)} = 1$, pour $i = 1, \ldots, m$ sont les *pivots* de M. En d'autres termes, à gauche (même ligne) et en dessous (même colonne) d'un pivot, il n'y a que des 0, et les pivots forment une ligne brisée décroissante ; $g(i)$ donne la colonne du i-ème pivot (celui de la i-ème ligne).

Exemple : ici $g(1) = 2$, $g(2) = 3$, $g(3) = 5$.

$$\begin{pmatrix} 0 & 1 & \times & \times & \times \\ 0 & 0 & 1 & \times & \times \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Théorème 2. *Les lignes d'une matrice échelonnée en lignes forment une famille libre.*

Algorithme . Noyau d'une application linéaire φ (version partielle)

Entrées: $M = (M_{i,j}) \in \mathcal{M}_{m\times n}(K)$ la matrice $\mathrm{Mat}(\varphi, \mathcal{B}, \mathcal{B}')$.
Sorties: Une matrice échelonnée en lignes qui a le même noyau que M.

Poser $i \leftarrow 0$.
pour $j = 1, 2, \ldots, m$ **faire** $\{$*on traite la colonne* $j\}$
  si il existe $p > i$, tel que $M_{p,j} \neq 0$ **alors** $\{$*un pivot sur cette colonne ?*$\}$
    $\{$*Comme* $p > i$, *la ligne* p *ne contient pas déjà un pivot*$\}$
    Choisir p tel que $M_{p,j} \neq 0$ comme *pivot*. Par exemple, $|M_{p,j}|$ minimal.
    Poser $i \leftarrow i + 1$.
    Échanger les lignes $M_{p,\bullet}$ et $M_{i,\bullet}$.
    Diviser la ligne $M_{j,\bullet}$ par $M_{i,j}$. $\{$*le pivot* $M_{i,j}$ *est en place,* $g(i) = j\}$
    pour $k = i + 1, \ldots, m$ **faire** $\{$*annuler* $M_{k,j}$ *en utilisant* $M_{i,j} = 1\}$
      $M_{k,\bullet} \leftarrow M_{k,\bullet} - M_{k,j} \times M_{i,\bullet}$
Retourner les i premières lignes de M. Le noyau est de dimension $n - i$.

Comme $p > j$ lors du choix du pivot, la ligne p ne contient pas déjà un pivot. Jusqu'ici c'est l'exact pendant des opérations sur les colonnes utilisées pour calculer l'image. Si on veut expliciter l'obtention d'une base du noyau, en particulier distinguer variables liées et paramètres libres, il faut poursuivre :

Algorithme . Noyau d'une application linéaire φ

Entrées: Une matrice M échelonnée en lignes de même noyau que $\mathrm{Mat}(\varphi, \mathcal{B}, \mathcal{B}')$.
Sorties: Une base échelonnée en lignes de $\operatorname{Ker} \varphi$ (donnée dans la base $\mathcal{B}$).

pour $x = i, i - 1, \ldots, 2$ **faire**
  pour $h = 1, \ldots, x - 1$ **faire** $\{$*annuler* $M_{h,g(x)}$ *utilisant* $M_{x,g(x)} = 1\}$
    $M_{h,\bullet} \leftarrow M_{h,\bullet} - M_{h,g(x)} \times M_{x,\bullet}$
$\{$*Maintenant les colonnes des pivots ont tous leurs autres coefficients à* $0.\}$
Les vecteurs $(a_1, \ldots, a_n)$ éléments du noyau vérifient

$$-a_{g(x)} = M_{x,g(x)+1} a_{g(x)+1} + \cdots + M_{x,n} a_n, \quad \text{pour } x = 1, \ldots, i.$$

Les i variables $a_{g(1)}, \ldots, a_{g(i)}$ sont liées, les autres sont des paramètres libres : le noyau est de dimension $n - i$. Si on veut afficher explicitement une base (échelonnée en lignes) :
pour $y = 1, \ldots, n$ **faire**
  si y n'est pas de la forme $g(x)$ **alors** $\{$*pas de pivot en colonne* $y\}$
    Afficher le vecteur $A = (a_1, \ldots, a_n)$ dont les coordonnées sont

$$a_j = \begin{cases} -1 & \text{si } j = y \\ M_{x,j} & \text{si } j = g(x) \\ 0 & \text{sinon.} \end{cases}$$

À la réflexion, mise sous forme échelonnée ligne et colonne sont essentiellement équivalentes : on passe de l'un à l'autre algorithme en travaillant sur la transposée de M. Il paraît naturel d'éviter de faire *deux* calculs très analogues si on a besoin de l'image *et* du noyau. On adapte le premier algorithme (image), qui fait des opérations élémentaires sur les colonnes[2]. L'idée est d'effectuer ces opérations sur M, en en gardant la trace sur une matrice auxiliaire U, grâce au principe suivant

Théorème 3. *Initialement* $U = \mathrm{Id}_n$, $M = A \in \mathcal{M}_{m\times n}(K)$. *Si on effectue la* même *suite d'opérations sur les* colonnes *de* M *et de* U, *alors on a toujours* $AU = M$. *La matrice* U *est toujours inversible.*

L'algorithme correspondant est plus compliqué que noyau/image, et on va en donner une forme inefficace. Le but est de mettre la matrice M sous une forme facilement utilisable. Pas seulement pour le calcul du noyau (échelonnée en lignes) ou de l'image (échelonnée en colonnes), pour lesquels ce qui précède est « optimal ».

On s'inspire du calcul de l'image, et on va écrire $MU = (H|0)$, où H est échelonnée en colonnes suivie de colonnes de 0.

Remarque 4. Ce n'est pas forcément assez général : pour répondre à une question sur M, il faudra répondre à une question sur U, qui n'a pas de forme particulièrement simple. En particulier, cette forme est adéquate pour le calcul de noyau et image, et la résolution de systèmes linéaires (en particulier le calcul de l'inverse quand M est carrée). Elle ne l'est pas pour le calcul du déterminant.

Pour résoudre le système linéaire $MX = Y$, d'inconnue X, on l'écrit

$$(MU)U^{-1}X = Y.$$

On résout $(H|0)Z = Y$, d'inconnue Z, qui est facile et on retourne $X = UZ$. C'est plus intéressant que la méthode classique dès que l'on veut résoudre plusieurs systèmes pour la même matrice M, mais en variant le second membre Y. Par exemple pour calculer M^{-1}.

[2] On pourrait très bien adapter l'autre, qui fait des opérations élémentaires sur les lignes.

Algorithme . « Simplification » d'une application linéaire φ (pour noyau, image, systèmes…)

Entrées: $M = (M_{i,j}) \in \mathcal{M}_{m\times n}(K)$ la matrice $\mathrm{Mat}(\varphi, \mathcal{B}, \mathcal{B}')$.

Sorties: Une matrice inversible U, une matrice échelonnée en colonnes H telle que $MU = (H|0)$.

$\{$*On met M sous forme échelonnée en colonnes. On garde trace des opérations sur les colonnes dans U. Si initialement $M = A$, on a toujours $AU = M$.*$\}$
Poser $j \leftarrow 0$, $U \leftarrow \mathrm{Id}_n$.
pour $i = 1, 2, \ldots, m$ **faire** $\{$*on traite la ligne i*$\}$
    si il existe $p > j$, tel que $M_{i,p} \neq 0$ **alors** $\{$*un pivot sur cette ligne ?*$\}$
        Choisir p tel que $M_{i,p} \neq 0$ comme *pivot*. Par exemple, $|M_{i,p}|$ minimal.
        Poser $j \leftarrow j + 1$.
        Échanger les colonnes $M_{\bullet,p}$ et $M_{\bullet,j}$, puis $U_{\bullet,p}$ et $U_{\bullet,j}$.
        Diviser les colonnes $M_{\bullet,j}$ et $U_{\bullet,j}$ par $M_{i,j}$ $\{$*le pivot $M_{i,j}$ est en place*$\}$
        pour $k = j + 1, \ldots, n$ **faire** $\{$*annuler $M_{i,k}$ en utilisant $M_{i,j} = 1$*$\}$
            $M_{\bullet,k} \leftarrow M_{\bullet,k} - M_{i,k} \times M_{\bullet,j}$
            $U_{\bullet,k} \leftarrow U_{\bullet,k} - U_{i,k} \times U_{\bullet,j}$
Retourner U et $H \leftarrow$ (les i premières colonnes de M). L'image est de dimension i, donnée par H ; le noyau est de dimension $n - i$, donné par les $n - i$ dernières colonnes de U.