Algorithme du pivot et opérations sur les sev de $\mathbb{Q}^n$

L'objectif de ce T.P. est d'implémenter les opérations usuelles sur les sous-espaces vectoriels de $\mathbb{Q}^n$ (base, dimension, calcul de la somme et de l'intersection de deux sev, comparaison pour l'inclusion). On représentera un sous-espace F de $\mathbb{Q}^n$ par une matrice $n \times p$ à coefficients rationnels dont les colonnes constituent une famille génératrice, non nécessairement libre, de F.

Les élèves travaillant en Maple ne doivent pas utiliser le paquetage `linalg`, ce serait tricher. Toutefois ils sont autorisés à importer les fonctions `rowdim` et `coldim` de ce paquetage (entrer `with(linalg,rowdim,coldim);` au début de la feuille de calcul) permettant d'obtenir simplement le nombre de lignes et de colonnes d'une matrice.

Les élèves travaillant en Caml devront utiliser la bibliothèque `num` qui implémente l'arithmétique exacte sur les nombres rationnels (entrer `#open "num";;` au début du programme). Les opérations usuelles sur les rationnels sont notées `+/`, `-/`, `*/` et `//`. Le symbole `=/` désigne le test d'égalité entre rationnels et la fonction `num_of_int` permet de convertir un entier ordinaire en rationnel de dénominateur 1.

1) Algorithme du pivot

a) Écrire une fonction `echelonne` qui prend en argument une matrice M et retourne une matrice M' échelonnée par rapport aux lignes et déduite de M par opérations sur les lignes uniquement. On utilisera l'algorithme du pivot vu en cours. On procédera à une réduction complète, c'est-à-dire que les pivots de M' seront mis à 1 et dans chaque colonne contenant un pivot tous les coefficients autres que le pivot seront nuls.

Exemple : en prenant la matrice $M = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 6 & 7 \end{pmatrix}$, on aboutit à $M' = \texttt{echelonne}(M) = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$.

Cette fonction `echelonne` devra procéder en premier lieu à une copie de la matrice passée en argument de façon à ne pas modifier cette matrice argument. On utilisera le code suivant

en Maple :

```
copie := proc(m::matrix)
local n, p, i, j, t;
    n := rowdim(m);
    p := coldim(m);
    t := matrix(n, p);
    for i from 1 to n do for j from 1 to p do t[i, j] := m[i, j] od od;
    eval(t)
end
```

en Caml :

```
let copie(m) =
  let n = vect_length(m)
  and p = vect_length(m.(0))        in
  let t = make_matrix n p m.(0).(0) in
  for i=0 to n-1 do for j=0 to p-1 do t.(i).(j) <- m.(i).(j) done done;
  t
;;
```

b) Écrire une fonction `pivots` qui prend en argument une matrice M échelonnée par rapport aux lignes et retourne un tableau contenant la liste des colonnes des pivots de M. Avec l'exemple précédent, on aura `pivots(echelonne(`M`))`$= (1 \quad 3)$

2) Base, dimension, somme, inclusion

A l'aide des fonctions précédentes écrire les fonctions :

– `base :` retourne une base d'un sev donné par une famille génératrice ;

- `dimension :` retourne la dimension d'un sev donné par une famille génératrice ;
- `somme :` retourne une base de $F+G$, F et G étant des sev donnés ;
- `inclus :` retourne un booléen indiquant si un sev F est inclus dans un sev G.

3) Noyau d'une application linéaire

Soit $f : \mathbb{Q}^p \longrightarrow \mathbb{Q}^n$ une application linéaire donnée par sa matrice M dans les bases canoniques de $\mathbb{Q}^p$ et $\mathbb{Q}^n$ et $F = \operatorname{Ker} f$. On veut déterminer une base de F. Soit $M' =$ `echelonne`(M). Supposons dans un premier temps que tous les pivots de M' sont situés dans la partie gauche de M'. Alors M' est de la forme

$$M' = \begin{pmatrix} I_r & U \\ 0 & 0 \end{pmatrix} \qquad (*)$$

où $r = \operatorname{rg}(M) = \operatorname{rg}(M')$, $U \in M_{r,p-r}(\mathbb{Q})$ et (on s'en convaincra) $X = \begin{pmatrix} U \\ -I_{p-r} \end{pmatrix} \in M_{p,p-r}(\mathbb{Q})$ représente une base de $\operatorname{Ker} f$. Dans le cas général il existe une permutation σ de $[\![1,p]\!]$ telle que la matrice $M'' = \begin{pmatrix} M'_{\sigma(1)}, \ldots, M'_{\sigma(p)} \end{pmatrix}$ soit de la forme $(*)$ où M'_i désigne la i-ème colonne de M'. Alors la matrice Y dont la i-ème ligne est la $\sigma(i)$-ème ligne de X représente une base de $\operatorname{Ker} f$.

Ainsi, pour déterminer une base de $\operatorname{Ker} f$, il suffit de procéder aux opérations suivantes :

a) Calculer M' ;
b) Calculer σ, où $\sigma(1), \ldots, \sigma(r)$ sont les numéros des colonnes contenant les pivots de M' et $\sigma(r+1), \ldots, \sigma(p)$ les numéros des autres colonnes ;
c) Construire la matrice Y (directement, sans construire X).

Programmer cela ... ; on pourra écrire une fonction `seconds`, analogue à la fonction `pivots`, qui retourne un tableau contenant la liste des colonnes de M ne comportant pas de pivots.

4) Orthogonal et intersection

A l'aide des fonctions précédentes écrire les fonctions :

- `orth :` retourne une base de l'orthogonal dans $(\mathbb{Q}^n)^*$ d'un sev F donné par une famille génératrice ;
- `intersect :` retourne une base de $F \cap G$, F et G étant des sev donnés.