

Les questions sont les bienvenues et peuvent être envoyées à nathanael.fijalkow@gmail.com.

Si vous utilisez OCaml, vous pouvez utiliser la bibliothèque `num` qui implémente l'arithmétique exacte sur les nombres rationnels (entrer `#load "nums.cma" ;; 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.

Si vous utilisez Caml light, vous pouvez utiliser des flottants.

PARTIE I : Polynôme caractéristique et interpolation de Lagrange

Soit $M \in \mathcal{M}_n(\mathbb{Q})$. Le polynôme caractéristique de M est $\chi_M = \det(M - XI_n) \in \mathbb{Q}_n[X]$. On étudie un algorithme de calcul de χ_M reposant sur le calcul d'un déterminant par la méthode du pivot de Gauss et sur la théorie de l'interpolation de Lagrange.

A - Calcul d'un déterminant par opérations élémentaires

On peut calculer $\det(M)$ en transformant M en une matrice triangulaire supérieure à l'aide d'opérations sur les lignes selon l'algorithme qui suit.

Traitement de la colonne j

Supposons avoir obtenu $M' = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix}$ avec $A \in \mathcal{M}_{j-1}(\mathbb{Q})$ triangulaire supérieure de diagonale $(1, \dots, 1)$ et $\lambda \in \mathbb{Q}$ tels que $\det(M) = \lambda \det(M') = \lambda \det(C)$ (on est dans ce cas initialement avec $j = 1$, $M' = M$ et $\lambda = 1$).

- Si $j = n$ alors $\det(M) = \lambda$.
- Sinon, si la première colonne de C est nulle (c'est-à-dire si $M'_{ij} = 0$ pour tout $i \geq j$) alors $\det(C) = 0$ et donc $\det(M) = 0$.
- Sinon, soit $M'_{ij} \neq 0$ avec $i \geq j$:
 1. si $i > j$, échanger les lignes i et j de M' et changer λ en $-\lambda$;
 2. diviser la ligne j de M' par M'_{jj} (maintenant non nul) et changer λ en $\lambda M'_{jj}$;
 3. pour tout $i > j$, retrancher M'_{ij} fois la ligne j de M' à la ligne i de M' ;
 4. passer alors à la colonne $j + 1$.

Programmer cet algorithme. On écrira une fonction `det` qui prend une matrice M comme argument et qui retourne son déterminant. Cette fonction devra procéder en premier lieu à une copie de la matrice M de façon à ne pas la modifier.

B - Calcul du polynôme caractéristique

On peut théoriquement calculer χ_M par la méthode précédente en remplaçant M par $M - XI_n$, mais ceci impose de travailler avec des matrices à coefficients dans le corps $\mathbb{Q}(X)$ et non dans \mathbb{Q} , ce qui est délicat. Une méthode plus efficace consiste à calculer $\chi_M(\lambda)$ dans \mathbb{Q} pour $n + 1$ valeurs rationnelles distinctes de λ et à en déduire les coefficients de χ_M par interpolation de Lagrange.

1. Interpolation de Lagrange

Soit $x_1, \dots, x_n, y_1, \dots, y_n \in \mathbb{Q}$ tels que les x_i soient deux à deux distincts et P_n l'unique polynôme de $\mathbb{Q}_{n-1}[X]$ tel que pour tout i , $P_n(x_i) = y_i$. On note $Q_n = (X - x_1) \dots (X - x_n)$ et on a les relations de récurrence :

$$\begin{cases} P_{n+1} = P_n + \frac{y_{n+1} - P_n(x_{n+1})}{Q_n(x_{n+1})} Q_n \\ Q_{n+1} = (X - x_{n+1}) Q_n \end{cases}$$

Écrire une fonction `lagrange` qui prend en argument deux tableaux x et y contenant respectivement les x_i et les y_i et qui retourne un tableau p contenant les coefficients du polynôme P_n (p est indexé par $\{0, \dots, n-1\}$ et $p(i)$ est le coefficient de X^i dans P_n). On programmera les calculs de $P_n(x_{n+1})$ et $Q_n(x_{n+1})$ *efficacement*.

2. Polynôme caractéristique

Écrire une fonction `poca` prenant une matrice M en argument et retournant son polynôme caractéristique, sous forme de tableau de coefficients.

PARTIE II : Méthode de Fadeev

On étudie ici une autre méthode permettant de calculer le polynôme caractéristique d'une matrice. Notons $\chi_M = \sum_{k=0}^n p_k X^k$ et $B = M - XI_n$.

1. Montrer que $\sum_{k=0}^{n-1} (k+1)p_{k+1}X^k = [\det(B)]' = \sum_{k=1}^n \det(B_k)$ où B_k est la matrice obtenue en dérivant la k -ième colonne de B .

2. En étudiant la transposée de la comatrice de la matrice B , qu'on notera C dans la suite, montrer que $[\det(B)]' = -\text{tr } C$.

3. Montrer qu'on peut écrire $C = \sum_{k=0}^{n-1} C_k X^k$ avec $C_k \in \mathcal{M}_n(\mathbb{Q})$. En déduire que pour tout $k \in \{0, \dots, n-1\}$, $(k+1)p_{k+1} = -\text{tr } C_k$.

4. En utilisant l'égalité reliant B à C , montrer les relations suivantes

$$\begin{cases} MC_0 = p_0 I_n \\ MC_k - C_{k-1} = p_k I_n \quad \forall k \in \{1, \dots, n-1\} \end{cases}$$

Quitte à poser C_n égale à la matrice nulle, on peut prolonger la dernière égalité pour $k = n$.

5. En déduire les relations de récurrence suivantes

$$\begin{cases} p_n = (-1)^n & C_n = 0_n \\ C_{k-1} = MC_k - p_k I_n & \text{et} \quad p_{k-1} = \frac{\text{tr}(MC_{k-1})}{n-k+1} \quad \forall k \in \{1, \dots, n\} \end{cases}$$

6. Écrire une fonction Caml implémentant cet algorithme de calcul des coefficients du polynôme caractéristique de la matrice M .

7. En remarquant que les dernières valeurs calculées par l'algorithme de Fadeev sont $C_0 = C(0) = {}^t \text{co}(M)$ et $p_0 = \det M$, en déduire un algorithme d'inversion de matrice, et l'écrire en Caml.

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

L'objectif 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 .

A - Algorithme du pivot

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 et on s'inspirera de l'algorithme de calcul du déterminant de la partie précédente. 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 partant de $M = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 6 & 7 \end{pmatrix}$, on aboutit à $\text{echelonne}(M) = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$.

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 3)`

B - Base, dimension, somme, inclusion

À 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 .

C - Noyau d'une application linéaire

Soit $f : \mathbb{Q}^p \rightarrow \mathbb{Q}^n$ une application linéaire donnée par sa matrice M dans les bases canoniques de \mathbb{Q}^p et \mathbb{Q}^n et $F = \text{Ker } f$. On veut déterminer une base de F . Soit $M' = \text{echelonne}(M)$. 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

$$M' = \begin{pmatrix} I_r & U \\ 0 & 0 \end{pmatrix} \tag{1}$$

où $r = \text{rg}(M) = \text{rg}(M')$, $U \in \mathcal{M}_{r,p-r}(\mathbb{Q})$ et (on s'en convaincra) $X = \begin{pmatrix} U \\ -I_{p-r} \end{pmatrix} \in \mathcal{M}_{p,p-r}(\mathbb{Q})$ représente une base de $\text{Ker } f$. Dans le cas général, il existe une permutation σ de $\{1, \dots, p\}$ telle que la matrice $M'' = (M'_{\sigma(1)} \cdots M'_{\sigma(p)})$ soit de la forme (1) 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 $\text{Ker } f$.

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

1. Calculer M' ;
2. Calculer σ , où $\sigma(1), \dots, \sigma(r)$ sont les numéros des colonnes contenant les pivots de M' et $\sigma(r+1), \dots, \sigma(p)$ les numéros des autres colonnes;
3. 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.

D - Orthogonal et intersection

À 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.

E - Tests

Testez vos fonctions, si ce n'est pas déjà fait, avec les matrices suivantes :

$$a = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 5 & 6 \\ 0 & 1 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad c = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 5 \\ 1 & 1 \end{pmatrix}$$

Faites les tests suivants et vérifier à la main les résultats :

```
dim a, dim b, dim c;;
base a, base b, base c;;
noyau a;;
somme a b;;
somme a c;;
inters a b;;
Array.map (Array.map string_of_num) (inters a c);;
orth a;;
orth b;;
orth c;;
```