Matrices

Dans ce TP, on veut programmer des matrices, en utilisant des listes de listes : une matrice sera représentée par la liste de ses lignes, chacune étant une liste de coefficients.

Exercice 1. Définir des fonctions `zero` et `identite` construisant les matrices nulles et identité (la première aura deux paramètres, la seconde un seul).

Exercice 2. Écrire une fonction `taille` donnant la taille d'une matrice, c'est-à-dire le couple (nombre de lignes, nombre de colonnes).

On veut maintenant programmer les différentes opérations sur les matrices. On prendra garde à vérifier dans les différentes fonctions que les opérations sont bien valides.

Exercice 3.
a) Écrire une fonction qui calcule le produit d'une matrice par un scalaire.
b) Écrire une fonction qui additionne deux matrices si les tailles sont compatibles.
c) Écrire une fonction qui calcule la transposée d'une matrice.
d) Écrire une fonction qui calcule le produit de deux matrices, si les tailles sont compatibles.

L'algorithme naïf de multiplication de matrices n'est pas très efficace ; on va en voir un autre, qui fonctionne pour les matrices carrées de taille une puissance de 2 : l'algorithme de Strassen.

Soient donc A, B et $C = AB$ trois matrices carrées de taille 2^n.

L'idée est la suivante : on divise nos matrices en quatre parties

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \text{ et } C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Volker Strassen a remarqué en 1969 qu'en posant

$$\begin{aligned} M_1 &= (A_{11} + A_{22})(B_{11} + B_{22}) & M_2 &= (A_{21} + A_{22})B_{11} \\ M_3 &= A_{11}(B_{12} - B_{22}) & M_4 &= A_{22}(B_{21} - B_{11}) \\ M_5 &= (A_{11} + A_{12})B_{22} & M_6 &= (A_{21} - A_{11})(B_{11} + B_{12}) \\ M_7 &= (A_{12} - A_{22})(B_{21} + B_{22}) \end{aligned}$$

on a alors

$$\begin{aligned} C_{11} &= M_1 + M_4 - M_5 + M_7 & C_{12} &= M_3 + M_5 \\ C_{21} &= M_2 + M_4 & C_{22} &= M_1 - M_2 + M_3 + M_6. \end{aligned}$$

Si l'algorithme naïf fait de l'ordre de p^3 opérations pour des matrices de taille p, l'algorithme de Strassen en fait de l'ordre de $p^{2,807}$.

Dans les questions qui suivent, on suppose que toutes les matrices passées en paramètre sont de taille une puissance de 2 (on ne le vérifiera pas).

Exercice 4.
a) Écrire une fonction qui, étant donnée une matrice A, renvoie la liste des quatre sous-matrices $A_{11}, A_{12}, A_{21}, A_{22}$.

b) Écrire la fonction inverse, c'est-à-dire qui calcule la matrice à l'aide des quatre sous-matrices.

c) Écrire une fonction `strassen` qui calcule le produit de deux matrices en utilisant l'algorithme de Strassen.

On va maintenant regarder l'affichage de matrices.

Exercice 5. a) Écrire une fonction naïve qui affiche une matrice. Que se passe-t-il pour la matrice $\begin{pmatrix} 123456789 & 1 \\ 0 & 1 \end{pmatrix}$?

b) Il faut donc adapter la fonction précédente. Écrire une fonction qui, étant donnée une matrice, renvoie la taille du coefficient le plus long (en tant que string).

c) Écrire une fonction qui prend en paramètre deux entiers n et p, et qui renvoie la chaîne de caractères qui est n avec des espaces supplémenaires tels que la longueur de la chaîne soit de longueur p.

d) Utiliser la fonction précédente pour afficher la matrice.

La dernière opération qu'on veut implémenter est le calcul du rang d'une matrice. On va donc devoir implémenter la méthode du pivot de Gauss. En échangeant des lignes, on choisira toujours le pivot de valeur absolue la plus grande. On ne fera aucune division dans les questions qui suivent.

Si $A = (a_{ij}) \in \mathcal{M}_{n,p}(\mathbb{R})$, on appellera *sous-matrice en k* la matrice des a_{ij} pour $k \leqslant i \leqslant n$ et $k \leqslant j \leqslant p$.

Exercice 6. a) Écrire une fonction qui, étant donnée une matrice et un entier k, renvoie le plus grand coefficient de la première colonne de la sous-matrice en k, ainsi que l'indice de la ligne où il se trouve.

b) Écrire une fonction qui, étant donnés une matrice et un entier k, échange la première ligne de la sous-matrice en k avec la ligne contenant le pivot maximum de la question précédente, puis élimine les coefficients de la première colonne de la sous-matrice en k à partir de la deuxième ligne.

c) En déduire une fonction inductive qui échelonne une matrice (on fera attention à ne pas modifier la matrice de départ).

d) En déduire une fonction qui calcule le rang d'une matrice.