Programme de colles : semaine 19, du 3/3 au 7/3

Les nouveautés par rapport à la semaine précédente sont en bleu.

1 Matrices

La notion de déterminant n'est au programme de BCPST que dans le cas de la dimension 2.

- on note $\mathcal{M}_{n,p}(\mathbb{K})$ l'ensemble des matrices à n lignes, p colonnes et à coefficients dans $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$
- matrices lignes, colonnes, carrées, nulle, triangulaires, diagonales. On note $\text{diag}(a_1, \dots, a_n)$ la matrice diagonale de coefficients $a_1, \dots, a_n \in \mathbb{K}$
- matrice identité, pour tout $A \in \mathcal{M}_{n,p}(\mathbb{K})$, $I_n A = A I_p = A$
- opérations matricielles : somme, multiplication par un scalaire, produit, puissance
- produit et puissance de matrices diagonales
- pièges du produit matriciel : en général $AB \neq BA$, et $\big(AB = AC \text{ et } A \neq 0\big) \not\Longrightarrow B = C$
- formule du binôme de Newton : si A et B commutent alors $\forall p \in \mathbb{N},\ (A+B)^p = \displaystyle\sum_{k=0}^{p} \binom{p}{k} A^k B^{p-k} = \sum_{k=0}^{p} \binom{p}{k} B^k A^{p-k}$
- exemples de calculs de puissances de matrices par récurrence.
- <u>matrices inversibles :</u> on note $GL_n(\mathbb{K})$ l'ensemble des matrices inversibles de taille n
  - propriétés de l'inverse : pour $A, B \in GL_n(\mathbb{K})$ et $\lambda \in \mathbb{K}^*$, $A^{-1}, \lambda A, AB \in GL_n(\mathbb{K})$ et $(A^{-1})^{-1} = A$, $(\lambda A)^{-1} = \frac{1}{\lambda} A^{-1}$, $(AB)^{-1} = B^{-1}A^{-1}$, inverse d'une matrice diagonale
  - exemple de calculs d'inverses en utilisant un polynôme annulateur *(aucun théorème général n'a été énoncé)*
  - inverse d'une matrice de taille 2 : pour $A = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathcal{M}_2(\mathbb{K})$ on définit $\det A = ad - bc$; A est inversible si et seulement si $\det A \neq 0$ et dans ce cas $A^{-1} = \dfrac{1}{\det A} \begin{pmatrix} d & -c \\ -b & a \end{pmatrix}$
- <u>matrice transposée</u> : on note A^T la transposée de A
  - opérations : pour A et B des matrices ad hoc et $\lambda \in \mathbb{K}$ on a : $(A+B)^T = A^T + B^T$, $(\lambda A)^T = \lambda A^T$, $(A^T)^T = A$, $(AB)^T = B^T A^T$, $(A^{-1})^T = (A^T)^{-1}$
  - matrices symétriques, antisymétriques, les matrices $A^T A$ et AA^T sont toujours symétriques
- <u>matrices et systèmes linéaires :</u>
  - écriture sous la forme $AX = Y$ d'un système linéaire
  - matrices échelonnées, on note $\text{rg}(A)$ le rang de A. Exemples de calculs du rang d'une matrice en fonction d'un paramètre dans ses coefficients
  - pour $A \in \mathcal{M}_{n,p}(\mathbb{K})$ on a : $\text{rg}(A) \leq \min(n,p)$, $\text{rg}(A) = \text{rg}(A^T)$.
  - $A \in \mathcal{M}_n(\mathbb{K})$ est inversible ssi $\text{rg}(A) = n$ ssi pour tout $Y \in \mathcal{M}_{n,1}(\mathbb{K})$ $AX = Y$ est de Cramer. Dans ce cas la solution de $AX = Y$ est $X = A^{-1}Y$
  - calcul pratique de l'inverse : par résolution du système $AX = Y$ pour Y quelconque, cela revient à faire des opérations sur les lignes des matrices A et I_n

2 Informatique en langage Python

- tableaux avec la bibliothèque `numpy` :
  - création avec les commandes : `np.array`, `np.ones`, `np.eye`, `np.zeros`, `np.diag`
  - opérations coefficients par coefficients. *Nous n'avons pas vu en cours le résultat de `A==B` si `A` et `B` sont des tableaux.*
  - accès aux dimensions avec `np.size(A,0)` et `np.size(A,1)`
  - accès ou modifications des coefficients : si `A` est un tableau à n lignes et p colonnes alors on accède à ses coefficients par la syntaxe `A[i,j]` pour $i \in [\![0, n-1]\!]$ et $j \in [\![0, p-1]\!]$
  - création d'une matrice à partir de la matrice nulle via deux boucles `for` imbriquées
- tri d'une liste de nombres : tri par sélection, tri par insertion. *Seules les versions "en place" ont été présentées en classe.*

3 Questions de cours

Cette semaine, les colles suivront le schéma suivant :

A) un calcul de primitives en utilisant la dérivée d'une composée, voir Feuille de remédiation 10, exo 2 page 4 : `https://cahier-de-prepa.fr/bcpst1b-berthelot/download?id=5713`

B) un calcul d'inverse d'une matrice de taille 3 sans paramètre en utilisant le pivot de Gauss

C) une ou deux questions de cours issue de la liste ci-dessous

D) exercices au choix du colleur

1. Donner la définition du produit matriciel.
2. Quand dit-on qu'une matrice est inversible ? Démontrer ensuite que $J = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ n'est pas inversible.
3. Démontrer que si $A, B \in GL_n(\mathbb{K})$ alors $AB \in GL_n(\mathbb{K})$ et $(AB)^{-1} = B^{-1}A^{-1}$
4. Énoncer le théorème donnant l'inverse d'une matrice de $\mathcal{M}_2(\mathbb{R})$ et l'appliquer sur un exemple choisi par l'examinateur.
5. Démontrer que si $A \in \mathcal{M}_{n,p}(\mathbb{K}),\ B \in \mathcal{M}_{p,q}(\mathbb{K})$ et $C \in \mathcal{M}_{q,r}(\mathbb{K})$ alors $(AB)C = A(BC)$.
6. Énoncer la formule du binôme de Newton pour les matrices *(à l'attention des élèves : attention, citer les hypothèses adéquates est également un attendu de cette question).*
7. Démontrer que pour $A \in \mathcal{M}_{n,p}(\mathbb{K})$ et $B \in \mathcal{M}_{p,q}(\mathbb{K})$ on a $(AB)^T = B^T A^T$.
8. Écrire une fonction Python `produitmat` prenant en arguments deux matrices A et B et renvoyant leur produit matriciel. Votre fonction devra gérer le cas où les tailles des matrices A et B ne sont pas compatibles.
9. Écrire une fonction Python `transpose` prenant en argument une matrice A et renvoyant sa transposée.
10. Écrire une fonction Python `tri` réalisant le tri d'une liste de nombres selon (au choix de l'examinateur) l'algorithme du tri par sélection ou du tri par insertion. Vous illustrerez les différentes étapes de l'algorithme sur une liste simple de votre choix.

La question de cours est notée sur 10 points, le reste des exercices sur 10 autres points.