Semaine 25 : 13 au 17 mai 2024

A. Espaces vectoriels : cf semaine 24

B. Applications linéaires

Note aux interrogateurs : on étudie exclusivement les applications linéaires de $\mathbb{K}^p$ dans $\mathbb{K}^n$.

$*$ définition, forme linéaires, endomorphismes; Opérations: somme, combinaison linéaire; $f(O_p) = O_n$; $f(-x) = -f(x)$.
$*$ **Noyau d'une application linéaire:** définition; sev de $\mathbb{K}^p$; CNS d'injectivité.
$*$ **Matrices et applications linéaires:** matrice qui représente une application linéaire dans deux bases fixées;
$*$ **Image d'une application linéaire:** définition; sev de $\mathbb{K}^n$, CNS de surjectivité;
Soit $(u_1, \dots, u_p)$ une base de $\mathbb{K}^p$ (pas forcément la base canonique!). Alors, $Imf = Vect\,(f(u_1), \dots, f(u_p))$
$*$ **Rang d'une application linéaire :** $rgf = \dim(Imf)$ (ne dépend pas du choix de la base); théorème du rang.
$*$ **Matrices et applications linéaires:**
matrice qui représente une application linéaire dans deux bases fixées;
calcul matriciel de $f(x)\ \forall x \in \mathbb{K}^p$; (cohérence des bases)
matrice de la somme, de λf, composée de deux applications linéaires (cohérence des bases)
matrice de f^{-1}

Capacités exigibles :

1. Montrer qu'une application est linéaire.
2. former la matrice d'une application linéaire dans deux bases fixées (pas forcément canoniques)
3. calculer le noyau, l'image, le rang d'une application linéaire.
4. Utilisation de matrices pour la bijectivité, le calcul de f^{-1} et les opérations (polynômes d'endomorphismes par exemple).

C. Langage Python

Simulation d'expériences aléatoires , de marches aléatoires .

Déroulement de la colle :

La colle commence par une question d'informatique (langage `python`) parmi , puis passage aux exercices :

1. Un mobile se déplace aléatoirement le long d'un axe horizontal d'origine 0, sur des points à coordonnées entières, positives ou nulles.
   Les déplacements sont effectués selon le protocole suivant:

   • à l'instant 0, le mobile est sur l'origine O d'abscisse 0.

   • si, pour tout entier naturel n, le mobile se trouve à l'instant n sur le point d'abscisse k ($0 \leqslant k \leqslant n$), alors il sera à l'instant $n+1$, soit sur le point d'abscisse $k+1$ avec la probabilité $\dfrac{1}{3}$, soit sur le point O avec la probabilité $\dfrac{2}{3}$.
   Écrire un programme qui simule cette marche aléatoire et renvoie l'abscisse du mobile à l'instant n.

2. On dispose d'une pièce et d'un pion placé à l'origine d'un axe gradué. On lance n fois la pièce. A chaque lancer, on déplace le pion d'une unité vers la gauche si on obtient pile, d'une unité vers la droite si on obtient face.
   On note X_n l'abscisse du pion à la fin de l'expérience.
   Écrire une fonction qui simule l'expérience aléatoire et renvoie l'abscisse du pion à la fin de l'expérience.

3. On étudie le mouvement aléatoire d'une puce qui se déplace sur les sommets d'un triangle ABC.
   A l'instant 0, la puce est en A et se déplace selon les règles suivantes:

   • si à l'instant n, la puce est en A, alors à l'instant $n+1$, elle passe en B avec la probabilité $\dfrac{1}{3}$ et en C avec la probabilité $\dfrac{2}{3}$.

   • si à l'instant n, la puce est en B, alors à l'instant $n+1$, elle passe en A ou C de façon équiprobable.

   • si à l'instant n, la puce est en C, alors à l'instant $n+1$, elle y reste.

   Écrire une fonction `marche(n)` qui simule cette marche aléatoire et qui renvoie la position de la puce à l'instant n.
   On note l'événement A_n ="la puce est en A à l'instant n".
   Écrire une fonction `proba(n,m)` qui renvoie une valeur approchée de $P(A_n)$. (m est le nombre de fois que l'on réalise l'expérience.)