ECG2B
Lycée Marcelin Berthelot

Année scolaire 2024-2025
Mathématiques

# Programme de colle n°12

## Semaine du 06/01

## Réduction des matrices
## Application aux chaînes de Markov et aux systèmes différentiels

**Pour cette colle, ce qui tient lieu d'« exercice étoilé » est de savoir résoudre un système différentiel pour une matrice $A \in \mathscr{M}_n(\mathbb{R})$ diagonalisable, avec $n$ pas trop grand.**

### Reprise du programme de réduction des matrices.

### Chaînes de Markov

- Rappels d'ECG1 sur les graphes : sommets, arêtes, graphes orientés, graphes pondérés.
- Matrice d'adjacence d'un graphe (orienté ou non).
- Graphe probabiliste : c'est un graphe
  - orienté et pondéré ;
  - pour tous $(i, j) \in [\![1, r]\!]^2$, on a au plus une arête $i \to j$ ;
  - pour tout $i \in [\![1, r]\!]$, la somme des poids des arêtes sortant du sommet $i$ est 1.

  On notera $r$ le nombre de sommets du graphe probabiliste considéré.
- Matrice de transition d'un graphe probabiliste.
- Définition : matrice stochastique. Une matrice est stochastique ssi c'est la matrice de transition d'un graphe probabiliste. Si M est stochastique, $1 \in \mathrm{Sp}(\mathrm{M})$ et $\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathrm{E}_1(\mathrm{M})$.
- Chaîne de Markov associée à un graphe probabiliste de matrice de transition M : c'est une suite de variables aléatoires $(\mathrm{X}_n)_{n\in\mathbb{N}}$, à valeurs dans $[\![1, r]\!]$, telles que

  $$\forall\,(i, j) \in [\![1, r]\!]^2,\ \mathbb{P}_{(\mathrm{X}_n=i)}(\mathrm{X}_{n+1} = j) = m_{i,j}$$

  où $m_{i,j}$ est le coefficient $(i, j)$ de M (c'est donc le poids de l'arête $i \to j$).
  Interprétation : position au temps $n$ d'un système sans mémoire « se déplaçant aléatoirement » sur le graphe en temps discret.
- On range la loi de $\mathrm{X}_n$ dans un vecteur de $\mathscr{M}_{1,r}^r$ :

  $$\mathrm{V}_n = \begin{pmatrix} \mathbb{P}(\mathrm{X}_n = 1) & \dots & \mathbb{P}(\mathrm{X}_n = r) \end{pmatrix}$$

  On a alors la relation $\mathrm{V}_{n+1} = \mathrm{V}_n\mathrm{M}$. $\mathrm{V}_n$ est appelé *$n$-ième état probabiliste de la chaîne.*
- $\forall\, n \in \mathbb{N},\ \mathrm{V}_n = \mathrm{V}_0\mathrm{M}^n$.
- État probabiliste stable : c'est un état probabiliste $\mathrm{V} = \begin{pmatrix} p_1 & \dots & p_r \end{pmatrix}$ (les $p_i$ sont donc positifs et de somme 1) tel que $\mathrm{VM} = \mathrm{V}$.

  Un état $\mathrm{V} = \begin{pmatrix} p_1 & \dots & p_r \end{pmatrix}$ est stable ssi $\begin{pmatrix} p_1 \\ \vdots \\ p_r \end{pmatrix}$ est vecteur propre de ${}^t\mathrm{M}$ pour la valeur propre 1.

## Systèmes différentiels linéaires

- Rappels d'ECG1 : résolution, pour $(a,b) \in \mathbb{R}^2$, de $y' + ay = 0$, $y'' + ay' + by = 0$ (lorsque l'équation caractéristique a au moins une racine réelle).
- Résolution d'équations non homogènes (la recherche d'une solution particulière doit être guidée).
- Prise en compte d'une condition initiale.
- Systèmes différentiels $X' = AX$ où $A \in \mathscr{M}_n(\mathbb{R})$ est diagonalisable.
  Résolution par diagonalisation de A, en posant $X = PY$ ; ou en écrivant $X(t) = \sum_{i=1}^{n} C_i e^{\lambda_i} t$ où $(C_i)$ est une base de vecteurs propres.
- Théorème de Cauchy : existence et unicité d'une solution à un problème de Cauchy (cas scalaire d'ordre 1 ou d'ordre 2 ; cas vectoriel)
- Quelques aspects qualitatifs :
  - trajectoire (l'ensemble des $(x_1(t), \ldots, x_n(t))$ pour $t \in \mathbb{R}$
  - trajectoire convergente (les $x_i$ ont une limite finie pour $t \to +\infty$)
  - point d'équilibre (ce sont les solutions constantes).

  Résultats associés : pour le système $(S)\ :\ X' = AX$ :
  - Les points d'équilibre de (S) sont les éléments de Ker (A).
  - Si $\mathrm{Sp}(A) \subset \mathbb{R}_-$, toute trajectoire converge vers un point d'équilibre.
  - Si $\mathrm{Sp}(A) \subset \mathbb{R}_-^*$, toute trajectoire converge vers 0.
  - Si A admet une vap > 0 il existe des solutions divergentes.

## Python

- Simulation d'expériences aléatoires : pas de commandes spécifiques aux chaînes de Markov mais on peut demander de simuler des expériences aléatoires simples avec `rd.random`, `rd.randint` par exemple.
- Algèbre linéaire : on doit savoir utilise les outils de `numpy.linalg`, notamment sur la matrice de transition. Ci-dessous les principales fonctions qui peuvent être utilisées :
  - `al.eig` : recherche d'éléments propres.
  - `al.rank` : rang d'une matrice.
  - `al.matrix_power` : mise à la puissance $n$ d'une matrice.
  - `np.dot` : produit matriciel.
  - `np.eye` : matrice identité.
  - `np.transpose` : transposition
- Systèmes différentiels : rien n'a été vu en Python à ce sujet pour l'instant.