Semaine n°23 du 08 au 12 avril

Informatique(Python) : cf exemples en annexe

- Tableau 1D, bibliothèque `Numpy`, fonction `array, len`

Suites réelles

- Suite majorée, minorée, bornée, (strictement) croissante/décroissante, (strictement) monotone, constante, stationnaire.
- Définition d'une suite $(u_n)_{n\in N}$ convergente de limite ℓ : $\forall \epsilon > 0, \exists n_\epsilon \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq n_\epsilon \Rightarrow |u_n - \ell| \leq \epsilon$
- Unicité de la limite d'une suite convergente, toute suite convergente est bornée(réciproque fausse)
- Suite divergente : définition.
- Limites des suites usuelles, opérations sur les limites, croissances comparées.
- Théorème de composition des limites : si $(u_n)_{n\in N}$ est convergente de limite ℓ et si f admet une limite L en ℓ, alors $(f(u_n))_{n\in N}$ est convergente de limite L.
- Si une suite est convergente vers une limite $l > 0$, alors à partir d'un certain rang les termes de la suites sont tous strictement positifs.
- Théorème de passage à la limite dans une inégalité, théorèmes de comparaison, théorème des gendarmes.
- Suites extraites $(u_{2n})_{n\in\mathbb{N}}$ et $(u_{2n+1})_{n\in\mathbb{N}}$. Utilisation pour montrer la convergence ou la divergence de $(u_n)_{n\in\mathbb{N}}$.
- Suites adjacentes : définition, propriété.
- suites équivalentes : définition, caractérisation en pratique, transitivité, produit, quotient, puissance, multiplication par un scalaire non nul. Deux suites équivalentes admettent les mêmes limites.
- Equivalents usuels : polynômes, et si (u_n) converge vers 0 :
  - $\sin(u_n) \underset{n\to+\infty}{\sim} u_n$
  - $\tan(u_n) \underset{n\to+\infty}{\sim} u_n$
  - $\ln(1+u_n) \underset{n\to+\infty}{\sim} u_n$
  - $e^{u_n} - 1 \underset{n\to+\infty}{\sim} u_n$
  - pour tout $\alpha \in \mathbb{R}^*$, $(1+u_n)^\alpha - 1 \underset{n\to+\infty}{\sim} \alpha u_n$ et en particulier (pour $\alpha = \frac{1}{2}$), $\sqrt{1+u_n} - 1 \underset{n\to+\infty}{\sim} \frac{u_n}{2}$
  - $1 - \cos(u_n) \underset{n\to+\infty}{\sim} \frac{u_n^2}{2}$
- Exemples d'études guidées de suites de type $u_{n+1} = f(u_n)$

Matrices

- Définitions : matrice nulle, carrée, identité, ligne, colonne, diagonale, triangulaire supérieure ou inférieure.
- Opérations : additions de deux matrices, multilpication par un scalaire, produit de matrices.
- Propriétés du produit : produit avec la matrice identité ou la matrice nulle, associativité, distributivité, non commutativité, non intégrité, $AB = AC \not\Rightarrow B = C$.
- Ecriture matricielle d'un système linéaire.
- Rang d'une matrice (= rang de son système associé), méthode de calcul du rang en échelonnant la matrice.
- Puissances de matrices carrées, cas particulier des matrices diagonales, polynômes de matrices.
- Binôme de Newton quand les matrices sont inversibles.

- Transposée d'une matrice : définition, propriétés, matrices symétriques et antisymétriques.
- Matrice carrée inversible : définition, propriétés, puissances avec exposant négatif.
- Recherche pratique de l'inverse : A est inversible si et seulement si son système associé $AX = B$ est un système de Cramer, détermination de l'inverse en résolvant le système en question.
- Critère d'inversibilité avec le rang.
- Recherche de l'inverse à l'aide d'un polynôme annulateur.
- critère d'inversabilité pour les matrices de taille 2 : déterminant. (Pas de formule donnant directement l'inverse.)

Statistiques univariée

- Définitions :Série statistiques, médiane, quartiles, écart interquartiles, moyenne (linéarité de la moyenne Démonstration exigible).
- Variance ; Définition, théorème de Koenig-huygens Démonstration exigible,
  $V(ax_i + b) = a^2V(x_i)$Démonstration exigible
- Ecart-type.

Remarques aux colleurs

— Merci aussi de poser une petite question d'informatique (cf Annexe).

Exemples de programmes informatiques

Exercice 1

Ecrire en Python une fonction `existence` qui prend en entrée un tableau 1D T et un nombre *element* et renvoie `True` si *element* se trouve dans le tableau T, `False` sinon.

```
def existence(T,element):
    a=len(T)      # nombre de lignes
    for i in range(a):      # parcours des lignes
            if T[i]==element:  # on teste si T[i] est égal à élément
                return True
    return False   # si on n' a pas trouvé element après avoir parcouru tout le tableau
```

Exercice 2

Ecrire en Python une fonction `MaximumTableau` qui prend en entrée un tableau 1D T et renvoie la plus grande valeur de ce tableau

```
def MaximumTableau(T):
    a=len(T)       # nombre de lignes
    maxi = T[0]  # initialisation avec la première valeur du tableau
    for i in range(a):        # parcours des lignes
            if T[i]>maxi:  # on teste si T[i] est plus grand
                maxi=T[i]    # on a trouvé une plus grande valeur
    return maxi
```

Exercice 3

Ecrire en Python une fonction `Moyenne` qui prend en entrée un tableau 1D T et renvoie la moyenne de ses éléments :

```
def Moyenne(T):
    a=len(T)      # nombre de lignes
    S=0           # initialisation de la somme
    for i in range(a):      # parcours des lignes
            S=S+T[i]      # on rajoute l'élément T[i]
    return S/a          # formule pour la moyenne
```