Semaine n°26 du 28 au 31 mai

Informatique(Python) : cf exemples en annexe

- Tableau 1D, bibliothèque `Numpy`, Représentation graphique.
- Simulation expérience aléatoire. Estimation des probabilités et de l'espérance.

Limites de fonctions

- Définitions avec les quantificateurs des limites suivantes :
  $\lim\limits_{x\to x_0} f(x)=\ell$, $\lim\limits_{x\to x_0} f(x)=+\infty$, $\lim\limits_{x\to x_0} f(x)=-\infty$, $\lim\limits_{x\to+\infty} f(x)=\ell$, $\lim\limits_{x\to+\infty} f(x)=+\infty$, ...
- Limite à gauche et limite à droite en un point, définition avec les quantificateurs.
- Asymptote verticale et horizontale.
- Opérations sur les limites : addition, multiplication, quotient, composition, formes indéterminées.
- Limites et inégalités : Si $\lim\limits_{x\to x_0} f(x)=l>0$ alors il existe un voisinage de x_0 sur lequel f est strictement positive, théorème des gendarmes, théorèmes de comparaison.
- Une fonction monotone sur un intervalle ouvert admet une limite en chacune des bornes de l'intervalle.
- Croissances comparées (logarithme, puissance,exponentielle)
- fonctions équivalentes : définition, équivalents usuels :
  - polynômes en $+\infty$, $-\infty$ et 0.
  - $\sin(x) \underset{x\to0}{\sim} x$
  - $\tan(x) \underset{x\to0}{\sim} x$
  - $\ln(1+x) \underset{x\to0}{\sim} x$
  - $e^x-1 \underset{x\to0}{\sim} x$
  - pour tout $\alpha\in\mathbb{R}^*$, $(1+x)^\alpha-1 \underset{x\to0}{\sim} \alpha x$ et en particulier (pour $\alpha=\frac{1}{2}$), $\sqrt{1+x}-1 \underset{x\to0}{\sim} \frac{x}{2}$
  - $1-\cos(x) \underset{x\to0}{\sim} \frac{x^2}{2}$
- Propriétés : transitivité, produit, quotient, puissance (pas de somme ni de composition !), théorème de substitution.

Continuité

- Définition de la continuité en un point, continuité à gauche et à droite.
- Continuité sur un intervalle, continuité des fonctions usuelles, opérations algébriques sur les fonctions continues, composition de fonctions continues.
- Prolongement par continuité.
- Théorème des valeurs intermédiaires.
- Image d'un intervalle par une fonction continue, image d'un segment par une fonction continue, une fonction continue sur un segment est bornée et atteint ses bornes.
- Théorème de la bijection, réciproque des fonctions puissances $x\mapsto x^n, n\in\mathbb{N}^*$ en distinguant suivant la parité de n, réciproque de la fonction tangente.

Applications linéaires

- Définition d'une application linéaire, image du vecteur nul, vocabulaire (endomorphisme, isomorphisme, automorphisme)

- Opérations : somme, multiplication par un scalaire, composition (démonstration exigible), réciproque d'applications linéaires.
- Noyau : définition, sous-espace vectoriel de l'ensemble de départ (démonstration exigible), caractérisation de l'injectivité,
- Image : définition, sous-espace vectoriel de l'ensemble d'arrivée, $Im(f) = vect(f(\vec{e_1}), ...f(\vec{e_p}))$ où $(\vec{e_1}), ...\vec{e_p}))$ base de l'ensemble de départ (démonstration exigible), caractérisation de la surjectivité.
- Détermination d'une application linéaire à partir de l'image d'une base, liens entre l'image d'une base et injection, surjection, bijection.

Remarques aux colleurs

— Merci aussi de poser une petite question d'informatique (cf Annexe).
— Pouvez vous vérifier la rédaction de l'étude des intervalles de continuité ?

Exemples de programmes informatiques

Exercice 1

Ecrire en Python une fonction `existence` qui prend en entrée un tableau T 1D 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
```