Programme de colles : semaine 13, du 6/1 au 10/1

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

1 Logique

- principe d'exemple et de contre-exemple pour prouver une propriété du type "$\exists\, x \in E \ : \mathcal{P}(x)$" ou nier une propriété du type "$\forall x \in E,\ \mathcal{P}(x)$"
- principe pour prouver une propriété du type "$\forall x \in E,\ \mathcal{P}(x)$" ("soit $x \in E$ alors ... donc $\mathcal{P}(x)$")
- principe de double implication pour prouver une équivalence
- principe de contraposition pour prouver une implication
- raisonnement par l'absurde
- raisonnement par analyse-synthèse

2 Ensembles

- vocabulaire : élément, partie, ensemble vide ; descriptions d'ensembles : en extension, par équation et par paramétrage
- inclusion, égalité, principe de double inclusion : savoir montrer que $A \subset B$, via "soit $x \in A$ alors ... donc $x \in B$". Savoir montrer que $A \not\subset B$
- égalité d'ensembles : savoir montrer que $A = B$ par principe de double inclusion, ou en montrant que $x \in A \iff x \in B$
- union et intersection : associativité, union et intersection finies, exemple d'union et d'intersection infinies
- complémentaire et différence : on note $\overline{A} = E \backslash A$ le complémentaire de A dans E, et par définition $A \backslash B = A \cap \overline{B}$
- formules de De Morgan
- distributivité de $\cap$ et $\cup$
- produit cartésien $I \times J$ de deux ensembles I et J, représentation dans $\mathbb{R}^2$ dans le cas où $I, J \subset \mathbb{R}$. Produit cartésien d'un nombre fini d'ensembles
- ensemble des parties $\mathcal{P}(E)$ d'un ensemble E
- majorants/minorants, maximum/minimum, bornes supérieure/inférieure d'une partie de $\mathbb{R}$. Toute partie non vide et majorée (resp. minorée) de $\mathbb{R}$ admet une borne supérieure (resp. inférieure). *Nous n'avons pas fait d'exercice sur ce point.*

3 Informatique en langage Python

Attention, les indices des éléments d'une liste n'ont pas été abordés en classe. Modification et suppression d'éléments seront également vus plus tard.

Listes : *Reprise du programme précédent.*
Bibliothèques :

- on recommande la syntaxe `import bibli as alias`
- bibliothèques `numpy` et `matplotlib.pyplot`
- savoir tracer le graphe d'une fonction ou d'une suite de nombres réels. *La fonction* `linspace` *n'a pas été abordée en classe. Les élèves doivent savoir définir une liste d'abscisses du type* `[a + k*(b-a)/N for k in range(N+1)]`

4 Questions de cours

Les premières minutes de la colle porteront sur une ou plusieurs des questions suivantes :

1. Si $\mathcal{P}$ et $\mathcal{Q}$ sont deux assertions, que sont la contraposée et la réciproque de $\mathcal{P} \implies \mathcal{Q}$?
2. Soit $n \in \mathbb{Z}$. Montrer que si n^2 est pair alors n est pair.
3. Montrer par analyse-synthèse que pour toute fonction $f : \mathbb{R} \longrightarrow \mathbb{R}$ il existe une unique fonction paire $f_1 : \mathbb{R} \longrightarrow \mathbb{R}$ et une unique fonction impaire $f_2 : \mathbb{R} \longrightarrow \mathbb{R}$ telles que $f = f_1 + f_2$.
4. Soient les ensembles $E = \{(t, 4t-1),\ t \in \mathbb{R}\}$ et $F = \{(x,y) \in \mathbb{R}^2 \mid 4x - y = 1\}$. Montrer que $E = F$.
5. Soient les ensembles $A = \{\left(\dfrac{1-t^2}{1+t^2}, \dfrac{2t}{1+t^2}\right),\ t \in \mathbb{R}\}$ et $B = \{(x,y) \in \mathbb{R}^2 \ :\ x^2 + y^2 = 1\}$. Montrer que $A \subsetneq B$ *(attention, ici il y a deux questions en une)*. On considérera l'élément $(-1, 0)$.
6. Soient A et B deux parties de E. Montrer que $A \subset B \iff \overline{B} \subset \overline{A}$.
7. Soit $K = \{(0,1), (1,0)\}$. Démontrer par l'absurde qu'il n'existe pas de parties A et B de $\mathbb{R}$ telle que $K = A \times B$.
8. Soit $A \subset \mathbb{R}$. Donner la définition de "A est majorée" ou de "M est un maajorant de A" ou de "s est la borne supérieure de A" ou des assertions similaires pour minorée/minorant/borne inférieure.
9. Écrire un programme Python permettant de tracer le graphe d'une fonction $f : [a,b] \longrightarrow \mathbb{R}$ choisie par l'examinateur.
10. Écrire un programme Python permettant de tracer u_n en fonction de n où (u_n) est une suite définie par récurrence choisie par l'examinateur.

La colle se poursuivra avec un ou plusieurs calcul "type remédiation" au sein ou non d'un exercice plus compliqué. Cette semaine, ces calculs doivent être similaires à ceux traités dans les feuilles de :

- Remédiation 8, exos 1, 2 et 3 : `https://cahier-de-prepa.fr/bcpst1b-berthelot/download?id=5559`

Attention : *les croissances comparées n'ont pas été vues en classe.*

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