

Remarques DS 7

À nouveau, les résultats globaux de ce DS sont un peu décevants. L'exercice sur les probabilités est plutôt bien réussi, mais les notions fondamentales de l'analyse (continuité, dérivabilité, limites) ne sont pas acquises pour la majorité des copies. L'exercice de cours et l'exercice d'informatique sont très hétérogènes.

Remarques importantes :

- Penser à repérer les questions faciles. Le sujet était long, il était donc judicieux de prendre quelques instants pour repérer les questions “faciles”. Trop de copies ont simplement “zappé” l'exercice 4, probablement sans même lire l'entièreté des questions. Dommage, car certaines questions étaient simples même sans avoir réussi les précédentes (par exemple, trouver $\lim_{n \rightarrow +\infty} v_n$ quand l'énoncé donne que $v_n \geq n \dots$).
- Ne faire que ce que demande l'énoncé. Dans l'exercice 2 question 2, de nombreuses copies ont fait l'étude complète de f alors qu'on demandait seulement de justifier qu'elle était continue... De même, en informatique, beaucoup de copies s'embarassent de boucles `for i in range(n)` et `for j in range(p)` où n et p sont les nombres de lignes et colonnes d'une matrice A ; l'énoncé précisait pourtant qu'on travaillait avec une matrice $A \in \mathcal{M}_2(\mathbb{R}) \dots$
- Réutiliser les questions précédentes. Dans l'exercice 5, la question 0) conduisait à une formule générale donnant $\mathbb{P}(F_3)$ en fonction de $\mathbb{P}(E_3)$. Dans les questions 1), 2) et 3) il fallait alors réutiliser cette formule pour obtenir $\mathbb{P}(F_3)$ à partir de $\mathbb{P}(E_3)$. Inutile donc de *refaire* le raisonnement de la question 0) à chacune des questions suivantes! Dans le même esprit, en informatique il faut *réutiliser* les fonctions précédentes et non recopier plusieurs fois le même code.

Remarques particulières :

- **CNS** : À la question 3) de l'exercice 3 on demandait une *condition nécessaire et suffisante* sur a et b pour que la matrice A soit inversible. En d'autres termes, on attendait une réponse donnant une *équivalence* du type : “ A est inversible si et seulement si a et b sont non nuls” ou “ A est inversible lorsque a et b sont non nuls”. Notez que les phrases du type “si a et b sont non nuls alors A est inversible” ou “pour que A soit inversible il faut que a et b soient non nuls” n'expriment qu'une *implication* et ne répondent donc pas à la question posée.
- **EDD** : Il est très étrange de lire qu'une fonction est définie sur un ensemble D et dérivable sur un ensemble D' avec $D' \not\subset D$! Comment une fonction pourrait-elle, par exemple, être définie sur $[-1, 1]$ mais dérivable sur $[-3, 3]$? Pour que f soit dérivable en x_0 (c'est-à-dire pour que $f'(x_0)$ existe), il faut (mais ne suffit pas) que f soit définie en x_0 (c'est-à-dire que $f(x_0)$ existe).
- **IFBO** : en informatique, prenez garde à la manipulation des booléens avec les tests `if`. Je rappelle à nouveau qu'écrire :
`if bool :`
 `return True`
`else :`
 `return False`
est particulièrement maladroit car ces 4 lignes se résument à : `return bool`. De même écrire `if bool == True` revient à écrire : `if bool`. Pensez également à utiliser les opérateurs logiques `and`, `not`, `or` pour construire vos booléens. Je rappelle enfin que le test “différent de” s'écrit `!=` en Python.
- Pour les quelques élèves concernés : arrêtez vos “blagues” sur les noms de fonctions Python. Le jour du concours, un nom de fonction “humoristique” serait non seulement très mal vu, mais pourrait en plus être interprété comme un signe distinctif que vous auriez volontairement inséré dans votre copie pour permettre à un correcteur de l'identifier malgré l'anonymisation : en d'autres termes, une tentative de triche...