

# Devoir Sur Table n°3 – Durée : 4h

L'utilisation de la calculatrice, des feuilles/notes de cours ou d'exercices est interdite.

La présentation, la rédaction, la clarté et la précision des raisonnements entreront dans l'appréciation de la copie.

Les résultats non encadrés/soulignés/surlignés ne seront pas pris en compte.

## Exercice 1 : Etude d'une suite implicite

Pour  $n \in \mathbb{N}^*$ , on considère la fonction  $f_n : x \mapsto \frac{x}{x-1}(x^n - 1)$ .

1. Soit  $n \in \mathbb{N}^*$  fixé.
  - (a) Donner le domaine de définition de  $f_n$  et justifier qu'elle y est continue.
  - (b) Montrer que  $f_n$  est prolongeable par continuité sur  $\mathbb{R}$ . On continuera de noter  $f_n$  ce prolongement. Préciser ainsi la nouvelle définition de la fonction  $f_n$  sur  $\mathbb{R}$  tout entier.
2. Définir une fonction Python nommée **f** qui prend en entrée un entier  $n \in \mathbb{N}^*$  et un réel  $x \in \mathbb{R}$  et renvoie la valeur de  $f_n(x)$ . On distingue les cas selon la valeur de  $x$ .
3. (a) Justifier que pour tout  $n \in \mathbb{N}^*$  et  $x \in \mathbb{R}$ ,  $f_n(x) = \sum_{k=1}^n x^k$ .
  - (b) Proposer alors une nouvelle façon de définir la fonction **f** en Python (cf. question 2.)
4. (a) Montrer que pour tout  $n \in \mathbb{N}^*$ , l'équation  $f_n(x) = 1$  admet une unique solution dans  $[0, 1]$ , que l'on notera  $u_n$ .
  - (b) On donne le programme Python suivant (la fonction **f** ayant été définie précédemment) :

```
def suite(n) :
    x = 0
    while f(n,x) < 1 :
        x += 0.001
    return x
```

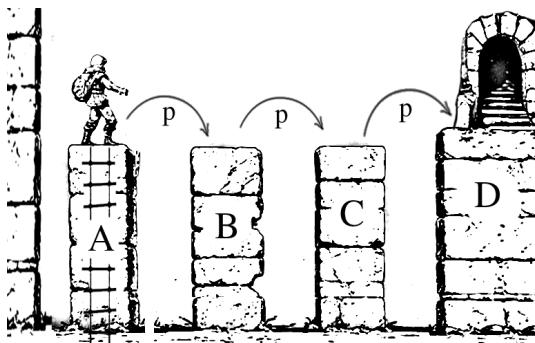
Expliquer ce programme. Quel est le lien entre la valeur renvoyée par la fonction **suite** et  $u_n$  ?

5. Montrer que pour tout  $n \in \mathbb{N}^*$ ,  $f_{n+1}(u_n) \geq 1$  et en déduire que la suite  $(u_n)_{n \in \mathbb{N}^*}$  est décroissante.
6. (a) Déterminer la valeur de  $u_2$  et vérifier que  $0 \leq u_2 < 1$ .
  - (b) Déduire que  $\lim_{n \rightarrow +\infty} (u_n)^n = 0$ .
7. Montrer finalement que  $(u_n)_{n \in \mathbb{N}^*}$  converge vers une valeur que l'on déterminera.

# Problème : Escape the dungeon !

Un aventurier cherche à s'échapper du donjon dans lequel on l'a enfermé.

À l'instant initial  $n = 0$ , il se situe sur la plateforme A, et doit effectuer 3 sauts consécutifs pour atteindre la sortie sur la plateforme D. On fixe  $p \in ]0, 1[$ . Chaque saut est indépendant et a une probabilité  $p$  de réussite :



- Si un saut est réussi, l'aventurier rejoint la prochaine plateforme à l'instant suivant.
- Si un saut est raté, l'aventurier tombe au sol et doit remonter à l'échelle. À l'instant suivant, il revient ainsi sur la première plateforme A et doit tout recommencer.
- Si l'aventurier atteint la plateforme D à l'instant  $n$ , il rejoint la sortie du donjon. Par convention, on décide alors qu'il se situe en D pour tous les instants suivants.

## Partie I : Relations de récurrence

Pour tout  $n \in \mathbb{N}$ , on note  $A_n$  l'évènement : "L'aventurier se situe sur la plateforme A à l'instant  $n$ ". On introduit, de même, les évènements  $B_n$ ,  $C_n$  et  $D_n$  pour les plateformes B, C et D.

1. (a) Donner les probabilités  $P(A_n)$ ,  $P(B_n)$ ,  $P(C_n)$ ,  $P(D_n)$  aux instants  $n = 0$ ,  $n = 1$  et  $n = 2$ .  
(b) Pour tout  $n \in \mathbb{N}$ , que vaut  $P(A_n) + P(B_n) + P(C_n) + P(D_n)$ ? Justifier.
2. (a) Pour tout  $n \in \mathbb{N}$ , montrer que  $P(A_{n+1}) = (1-p)P(A_n) + (1-p)P(B_n) + (1-p)P(C_n)$ .  
(b) Pour tout  $n \in \mathbb{N}$ , montrer que  $P(B_{n+1}) = p \times P(A_n)$  et exprimer similairement les probabilités  $P(C_{n+1})$  et  $P(D_{n+1})$  en fonction de  $P(A_n)$ ,  $P(B_n)$ ,  $P(C_n)$ ,  $P(D_n)$  et  $p$ .  
(c) En déduire la relation de récurrence triple suivante :

$$\forall n \in \mathbb{N}, P(A_{n+3}) = (1-p)(P(A_{n+2}) + pP(A_{n+1}) + p^2P(A_n)).$$

3. Compléter la fonction Python suivante pour que, à la donnée de  $p \in ]0, 1[$  et  $n \in \mathbb{N}$ , elle calcule et renvoie la valeur de la probabilité  $P(A_n)$ . **On copiera le programme en entier.**

```
def proba(p,n) :  
    u = 1 ; v = ... ; w = ...  
    for k in range(...):  
        x = ...  
        u = v  
        v = w  
        w = x  
    return ...
```

## Partie II : Recherche de racine et conséquences

On rappelle que  $p \in ]0, 1[$  est une valeur fixée.

On cherche à montrer que l'équation  $(E) : x^3 = (1-p)(x^2 + px + p^2)$  admet une unique solution dans  $\mathbb{R}$ .

4. Justifier que  $(E)$  admet au moins une solution dans  $]0, 1[$ .
5. Pour quelle valeur de  $p \in ]0, 1[$  le réel  $x = p$  est-il une solution de l'équation  $(E)$ ?
6. Montrer que pour tout réel  $x \neq p$ , l'égalité  $(E)$  est équivalente à  $(E') : x^3(1-x) = p^3(1-p)$ .  
*Indication : Multiplier chaque côté de  $(E)$  par  $(x-p)$ .*
7. (a) Justifier que toutes les solutions de  $(E')$  sont dans  $]0, 1[$ .  
(b) Dresser le tableau de variation complet de la fonction  $\varphi : x \mapsto x^3(1-x)$  sur  $]0, 1[$ . Déduire que :
  - Si  $p = \frac{3}{4}$ , l'équation  $(E')$  admet  $p$  pour unique solution.
  - Si  $p \neq \frac{3}{4}$ , l'équation  $(E')$  admet exactement deux solutions :  $p$ , et une autre valeur dans  $]0, 1[$ .
8. En distinguant les cas  $p = \frac{3}{4}$  et  $p \neq \frac{3}{4}$ , conclure que  $(E)$  admet toujours une unique solution dans  $\mathbb{R}$ .

Dans toute la suite, on note  $\alpha \in ]0, 1[$  l'unique solution de l'équation  $(E)$ .

9. (a) Lorsque  $p \in ]0, 1[$  et  $\varepsilon > 0$  sont des valeurs spécifiées par l'utilisateur, on souhaite calculer à l'aide de Python une valeur approchée de  $\alpha$ , solution de  $(E)$ , à  $\varepsilon$  près. Compléter, à cette fin, le script suivant. On pourra, au préalable, définir une certaine fonction  $f$ , qui sera exploitée dans la fonction dichotomie. **On copiera le programme en entier.**

```
def dichotomie(p, eps) :
    a = ... ; b = ...
    while ..... :
        c = (a+b)/2
        if ..... :
            a = ....
        else :
            b = ....
    return a
```

- (b) Dans le cas où  $p = \frac{1}{2}$  (seulement dans cette question) l'appel de `dichotomie(0.5, 0.001)` renvoie la valeur : 0.91894. Quel encadrement en déduit-on pour  $\alpha$  ?
10. On définit la constante  $C = \max \left( 1, \frac{p}{\alpha}, \frac{1-p}{\alpha^2} \right)$  (c'est-à-dire la plus grande de ces 3 valeurs). En raisonnant par récurrence triple, montrer que :  $\forall n \in \mathbb{N}, P(A_n) \leq C\alpha^n$ .
11. (a) En déduire  $\lim_{n \rightarrow +\infty} P(A_n)$ , puis  $\lim_{n \rightarrow +\infty} P(B_n)$  et  $\lim_{n \rightarrow +\infty} P(C_n)$ .  
(b) Etablir que  $\lim_{n \rightarrow +\infty} P(D_n) = 1$ . Comment peut-on interpréter ce résultat pour notre aventurier ?
- On note  $Z$  l'évènement "L'aventurier finit (à un moment) par s'échapper du donjon".
12. Pour tout  $n \in \mathbb{N}$ , comparer les évènements  $D_n$  et  $Z$  (au sens de l'inclusion). En déduire que  $P(Z) = 1$ .  
*On dit, dans ce genre de cas, que l'aventurier finira "presque-sûrement" par s'échapper du donjon.*

### Partie III : Nombre de tentatives

Pour finir, on segmente le parcours aléatoire de l'aventurier en différentes "tentatives" :

- L'aventurier démarre en A, et tente d'effectuer 3 sauts pour atteindre D : c'est la première tentative.
- Si l'aventurier tombe, il remonte sur la plateforme A et recommence : c'est une nouvelle tentative.

Pour tout  $k \in \mathbb{N}^*$ , on note  $T_k$  l'évènement : "L'aventurier atteint la sortie au bout de  $k$  tentatives exactement".  
Pour tout  $n \in \mathbb{N}^*$ , on note  $S_n$  l'évènement : "L'aventurier atteint la sortie en  $n$  tentatives ou moins".

13. Pour tout  $n \in \mathbb{N}^*$ , exprimer l'évènement  $S_n$  en fonction des évènements  $T_k$ .
14. A quelle situation correspond l'évènement  $T_1$ ? Justifier que  $P(T_1) = p^3$ .
15. (a) Pour tout  $k \geq 2$ , justifier que :

$$\bullet T_k \subset \bigcap_{i=1}^{k-1} \overline{T_i}, \quad \bullet P_{\overline{T_1} \cap \overline{T_2} \cap \dots \cap \overline{T_{k-1}}}(T_k) = p^3, \quad \bullet P_{\overline{T_1} \cap \overline{T_2} \cap \dots \cap \overline{T_{k-1}}}(\overline{T_k}) = 1 - p^3.$$

- (b) En déduire l'expression de  $P\left(\bigcap_{i=1}^k \overline{T_i}\right)$  en fonction de  $k \in \mathbb{N}^*$  et de  $p$ .  
(c) En déduire que pour tout  $n \in \mathbb{N}^*$ ,  $P(S_n) = 1 - (1 - p^3)^n$ .

16. (a) Pour tout  $k \in \mathbb{N}^*$ , montrer que  $P(T_k) = (1 - p^3)^{k-1} p^3$ .  
(b) En sommant ces probabilités, retrouver d'une autre façon la valeur de  $P(S_n)$  établie en 15.(c).
17. (a) Que vaut  $\lim_{n \rightarrow +\infty} P(S_n)$ ? Retrouver alors  $P(Z) = 1$ , où  $Z$  est l'évènement introduit en partie II.  
(b) Généralisation : Soit  $N \in \mathbb{N}^*$  un entier fixé. Au lieu de 3 sauts, on suppose que l'aventurier doit effectuer  $N$  sauts consécutifs pour rejoindre la plateforme D (la probabilité pour chaque saut étant toujours  $p$ ).  
Donner, en justifiant brièvement, la probabilité  $P(S_n)$  pour tout  $n \in \mathbb{N}^*$ .  
Déduire encore une fois que l'aventurier finira presque-sûrement par s'échapper du donjon.