Correction : DS 7

Exercice 1. Soit f la fonction définie par

$$f(x) = \frac{(x-1)\sin(x)}{x\ln(x)}$$

1. Déterminer l'ensemble de définition de f.
2. Déterminer les limites de f au bord de cet ensemble.
3. f est-elle prolongeable par continuité en 0 ?

Correction 1.

1. f est définie pour tout x tel que $\ln(x)$ soit défini et $x\ln(x) \neq 0$ c'est à dire pour $x > 0$ et $x \neq 1$.

$$\boxed{D_f =]0,1[\cup]1,+\infty[}$$

2. En $+\infty$: Pour tout $x \in \mathbb{R}$:

$$-1 \leq \sin(x) \leq 1$$

Donc pour tout $x > 1$

$$\frac{-(x-1)}{x\ln(x)} \leq f(x) \leq \frac{(x-1)}{x\ln(x)}$$

(attention au signe de $\ln(x)$ entre 0 et 1)
Or $\dfrac{(x-1)}{x\ln(x)} \sim_{+\infty} \dfrac{1}{\ln(x)}$ et $\lim\limits_{x\to+\infty} \dfrac{1}{\ln(x)} = 0$ Donc

$$\lim_{x\to+\infty} \frac{-(x-1)}{x\ln(x)} = \lim_{x\to+\infty} \frac{(x-1)}{x\ln(x)} = 0$$

Ainsi par le théorème d'encadrement, on a

$$\boxed{\lim_{x\to+\infty} f(x) = 0}$$

En 1 : On se propose de faire le changement de variable $y = x - 1$ On obtient

$$\lim_{x\to1} f(x) = \lim_{y\to0} \frac{y\sin(y+1)}{(y+1)\ln(y+1)}$$

Or

$$\ln(y+1) \sim_0 y$$

d'où

$$\frac{y\sin(y+1)}{(y+1)\ln(y+1)} \sim_0 \frac{\sin(y+1)}{y+1}$$

Ainsi

$$\boxed{\lim_{x\to1} f(x) = \sin(1)}$$

En 0

On a $\sin(x) \sim_0 x$ et $x - 1 \sim -1$ Donc

$$f(x) \sim \frac{-x}{x\ln(x)} \sim \frac{-1}{\ln(x)}$$

Or $\lim_{x\to 0} \ln(x) = -\infty$, donc

$$\boxed{\lim_{x\to 0} f(x) = 0}$$

3. $\boxed{\text{Ainsi } f \text{ est prolongeable par continuité en } 0}$

Exercice 2. 1. Montrer que $\displaystyle\int_0^1 \ln(x+1)dx = 2\ln(2) - 1$

2. Soit $P_n = \dfrac{1}{\sqrt{n}}\left(\displaystyle\prod_{k=1}^{n}(n+k)\right)^{\frac{1}{2n}}$. Montrer que pour otut $n \geq 1$ on a :

$$\ln(P_n) = \frac{1}{2n}\sum_{k=1}^{n} \ln\left(1 + \frac{k}{n}\right)$$

3. En déduire la valeur de la limite de P_n quand $n \to +\infty$.

Correction 2.

1. On fait une intégration par parties avec $u(x) = \ln(x+1)$ d'où $u'(x) = \frac{1}{x+1}$ et $v(x) = (x+1)$ une primitive de $v'(x) = 1$ On obtient

$$\begin{aligned}\int_0^1 \ln(x+1)dx &= [(x+1)\ln(x+1)]_0^1 - \int_0^1 1dx \\ &= 2\ln(2) - 1\ln(1) - [x]_0^1 \\ &= 2\ln 2 - 1\end{aligned}$$

2. Soit $P_n = \frac{1}{\sqrt{n}}\left(\displaystyle\prod_{k=1}^{n}(n+k)\right)^{\frac{1}{2n}}$ on a donc

$$\begin{aligned}\ln(P_n) &= \ln(\frac{1}{\sqrt{n}}) + \sum_{k=1}^{n}\frac{1}{2n}\ln(n+k) \\ &= \ln(\frac{1}{\sqrt{n}}) + \frac{1}{2n}\sum_{k=1}^{n}\ln\left(n(1+\frac{k}{n})\right) \\ &= -\frac{1}{2}\ln(n) + \frac{1}{2n}n\ln(n) + \frac{1}{2n}\sum_{k=1}^{n}\ln\left(1+\frac{k}{n}\right) \\ &= \frac{1}{2}\frac{1}{n}\sum_{k=1}^{n}\ln\left(1+\frac{k}{n}\right)\end{aligned}$$

Ainsi on reconnait une somme de Riemann et on obtient que

$$\begin{aligned}\lim_{n\to+\infty} \ln(P_n) &= \frac{1}{2}\int_0^1 \ln(1+x)dx \\ &= \ln(2) - \frac{1}{2}\end{aligned}$$

Et donc en revenant à P_n par la fonction exponentielle, continue sur $\mathbb{R}$, on a :

$$\boxed{\lim_{n\to+\infty} P_n = \exp(\ln(2) - \frac{1}{2}) = \frac{2}{\exp(1/2)}}$$

Exercice 3. On suppose que f est une fonction définie sur $[0,1]$ à valeurs dans $[0,1]$ et qu'il existe $k \in]0,1[$ tel que

$$\forall (x,y) \in [0,1]^2,\ |f(x) - f(y)| \leq k|x-y|.$$

Une telle fonction s'appelle une fonction k-contractante.

1. Montrer que f est continue sur $[0,1]$.
2. On appelle point fixe de f, un réel $x \in [0,1]$ tel que $f(x) = x$
   Déduire de la question précédente que f admet au moins un point fixe.
3. Montrer par l'absurde que ce point fixe est unique. On le note c dans le reste de l'énoncé.
4. On considère alors une suite $(c_n)_{n\in\mathbb{N}}$ définie par son premier terme $c_0 \in [0,1]$ et par la relation de récurrence : $\forall n \in \mathbb{N},\ c_{n+1} = f(c_n)$.
   (a) Montrer que pour tout $n \in \mathbb{N}$, $|c_n - c| \leq k^n |c_0 - c|$. (On rappelle que c désigne l'unique point fixe de f)
   (b) En déduire la limite de la suite $(c_n)_{n\in\mathbb{N}}$.

Correction 3.

1. • On cherche à étudier la continuité de f sur $[0,1]$. On repasse pour cela par la définition de la continuité en montrant que pour tout $x_0 \in [0,1]$, f est continue en x_0. Pour cela il faut donc montrer que pour tout $x_0 \in [0,1]$: $\lim\limits_{x\to x_0} f(x) = f(x_0)$.
   Soit donc $x_0 \in [0,1]$ fixé. On cherche donc à montrer que $f(x) - f(x_0)$ tend vers 0 lorsque x tend vers x_0. Mais par définition d'une fonction k-contractante, on sait que :

   $$\forall x \in [0,1],\ |f(x) - f(x_0)| \leq k|x - x_0|.$$

   On va donc obtenir le résultat voulu en utilisant le corollaire du théorème des gendarmes. En effet, on a :
   ⋆ $\lim\limits_{x\to x_0} k|x - x_0| = 0$ par propriété sur les somme, composée et produit de limites.
   ⋆ $\forall x \in [0,1],\ |f(x) - f(x_0)| \leq k|x - x_0|$.
   Ainsi d'après le corollaire du théorème des gendarmes, on a : $\lim\limits_{x\to x_0} f(x) - f(x_0) = 0 \Leftrightarrow \lim\limits_{x\to x_0} f(x) = f(x_0)$. Ainsi on a montré que la fonction f est continue en x_0 et comme cela est vraie pour tout $x_0 \in [0,1]$, on a la continuité de f sur $[0,1]$.
2. (TD 17 EX21.1)

3. Pour obtenir l'unicité du point fixe, on suppose par l'absurde qu'il existe deux points fixes $(c,d) \in [0,1]^2$ de f différents. Ainsi, on a : $|f(c)-f(d)| \leq k|c-d| \Leftrightarrow |c-d| \leq k|c-d|$. Or $0<k<1$ et ainsi, on a : $|c-d|<|c-d|$: absurde. Ainsi $c=d$ et f admet bien un unique point fixe.

4. (a)
   - On montre par récurrence sur $n \in \mathbb{N}$ la propriété $\mathcal{P}(n)$: $|c_n - c| \leq k^n |c_0 - c|$.
   - Initialisation : pour $n=0$: d'un côté, on a : $|c_0 - c|$ et de l'autre côté, on a : $k^0|c_0-c| = |c_0-c|$. Donc $\mathcal{P}(0)$ est vraie.
   - Hérédité : soit $n \in \mathbb{N}$ fixé, on suppose la propriété vraie au rang n, montrons que $\mathcal{P}(n+1)$ est vraie. D'après la définition de la fonction f, on sait que : $|f(c_n) - f(c)| \leq k|c_n - c| \Leftrightarrow |c_{n+1} - c| \leq k|c_n - c|$ car c est le point fixe de f. Puis par hypothèse de récurrence, on sait aussi que $|c_n - c| \leq k^n|c_0 - c|$. Ainsi comme $k>0$, on a : $k|c_n - c| \leq k^{n+1}|c_0 - c|$. Puis : $|c_{n+1} - c| \leq k^{n+1}|c_0 - c|$. Donc $\mathcal{P}(n+1)$ est vraie.
   - Conclusion : il résulte du principe de récurrence que pour tout $n \in \mathbb{N}$, on a : $|c_n - c| \leq k^n|c_0 - c|$.

   (b) On peut alors utiliser le corollaire du théorème des gendarmes et on obtient que :
   - $\forall n \in \mathbb{N},\ |c_n - c| \leq k^n|c_0 - c|$.
   - $\lim\limits_{n\to+\infty} k^n|c_0 - c| = 0$ car $-1<k<1$.

   Ainsi d'après le corollaire du théorème des gendarmes, on a : $\lim\limits_{n\to+\infty} c_n = c$.

Exercice 4. Le petit Tchoupi fait des nuits de longueur différentes. Il dort moins de 6 heures avec probabilité $p \in]0,1[$ et plus de 6 heures avec une probabilité $1-p$. En fonction de la durée de sommeil du petit Tchoupi, M G. fait parfois des erreurs de signes dans ses calculs... A chaque nouveau calcul, il a une probabilité $\frac{1}{10}$ de faire une faute si il a bien dormi et $\frac{1}{2}$ si la nuit a été courte... A une journée fixée, les calculs sont supposés indépendants entre eux. On note E_k l'événement { M G. fait une erreur au calcul k}.

1. Calculer, en fonction de p, $P(E_1)$.
2. M. G fait une faute à son premier calcul. Quelle est la probabilité qu'il ait bien dormi. (On exprimera le résultat en fonction de p)
3. Soit T l'événement { Tchoupi a bien dormi }. A-t-on $P_T(E_1 \cap E_2) = P_T(E_1)P_T(E_2)$? A-t-on $P(E_1 \cap E_2) = P(E_1)P(E_2)$? (Attention le fait d'être indépendant dépend fortement de la probabilité considérée, cette question doit être réfléchie...)
4. M. G fait une faute à son premier calcul. Quelle est la probabilité qu'il fasse une faute au deuxième calcul.

Correction 4.

1. Comme indiqué plus loin dans le sujet appelons T l'événement { Tchoupi a bien dormi }. Remarquons qu'il dort bien si il dort plus de 6h et qu'une conséquence directe est la bonne nuit de M. G. En appliquant la formule des probabilités totales aux SCE $(T, \overline{T})$ on obtient

$$\begin{aligned} P(E_1) &= P_T(E_1)P(T) + P_{\overline{T}}(E_1)P(\overline{T}) \\ &= \frac{1}{10}(1-p) + \frac{1}{2}p \\ &= \frac{1}{10} + \frac{2}{5}p \end{aligned}$$

2. On utilise la formule de Bayes on obtient :

$$\begin{aligned} P_{E_1}(T) &= \frac{P_T(E_1)P(T)}{P(E_1)} \\ &= \frac{\frac{1}{10}(1-p)}{\frac{1}{10} + \frac{2}{5}p} \\ &= \frac{1-p}{1+4p} \end{aligned}$$

3. L'énoncé nous annonce qu'à un jour donné les calculs sont indépendants entre eux. Donc on a $P_T(E_1 \cap E_2) = P_T(E_1)P_T(E_2)$. En revanche, le fait de faire une erreur n'est pas du tout indépendant du fait d'en faire une deuxieme... En effet si une premiere erreur a été commise, il y a beaucoup plus de chance d'en refaire une ensuite. Il n'y a donc aucune raison d'avoir $P(E_1 \cap E_2) = P(E_1)P(E_2)$
4. On cherche $P_{E_1}(E_2)$

$$\begin{aligned} P_{E_1}(E_2) &= \frac{P(E_1 \cap E_2)}{P(E_1)} \\ &= \frac{P_T(E_1 \cap E_2)P(T) + P_{\overline{T}}(E_1 \cap E_2)P(\overline{T})}{P(E_1)} \\ &= \frac{P_T(E_1)P_T(E_2)P(T) + P_{\overline{T}}(E_1)P_{\overline{T}}(E_2)P(\overline{T})}{P(E_1)} \\ &= \frac{\frac{1}{10^2}(1-p) + \frac{1}{2^2}p}{\frac{1}{10} + \frac{2}{5}p} \end{aligned}$$

Exercice 5. Roudoudou le hamster vit une vie paisible de hamster. Il a deux activités : manger et dormir... On va voir Roudoudou à 00h00 ($n = 0$). Il est en train de dormir.

— Quand Roudoudou dort à l'heure n, il y a 7 chances sur 10 qu'il dorme à l'heure suivante.

— Quand Roudoudou mange à l'heure n, il y a 2 chances sur 10 qu'il dorme à l'heure suivante.

On note D_n l'événement 'Roudoudou dort à l'heure n' et M_n 'Roudoudou mange à l'heure n'. On note $d_n = P(D_n)$ et $m_n = P(M_n)$ les probabilités respectives.

1. Justifier que $d_n + m_n = 1$.
2. Montrer rigoureusement que
$$d_{n+1} = 0,7d_n + 0,2m_n$$
3. Exprimer de manière similaire m_{n+1} en fonction de d_n et m_n.
4. Soit A la matrice
$$A = \frac{1}{10}\begin{pmatrix} 7 & 2 \\ 3 & 8 \end{pmatrix}.$$
et $X_n = \begin{pmatrix} d_n \\ m_n \end{pmatrix}$. Exprimer X_{n+1} en fonction de A et X_n. Puis en déduire - et la prouver- une expression de X_n en fonction de n, X_0 et A.
On admet que $\forall n \in \mathbb{N}$, $A^n = \frac{1}{5}\begin{pmatrix} 3\,(1/2)^n + 2 & -2\,(1/2)^n + 2 \\ -3\,(1/2)^n + 3 & 2\,(1/2)^n + 3 \end{pmatrix}$.
5. En déduire la valeur de d_n en fonction de n.

Correction 5.

1. D_n et M_n forment un système complet d'événements donc $d_n + m_n = 1$.
2. On cherche à calculer $d_{n+1} = P(D_{n+1})$ On applique la formule des probabilités totales avec le SCE (M_N, D_N)
$$\begin{aligned} d_{n+1} &= P(D_{n+1} \mid M_n)P(M_n) + P(D_{n+1} \mid D_n)P(D_n) \\ &= P(D_{n+1} \mid M_n)m_n + P(D_{n+1} \mid D_n)d_n \end{aligned}$$
L'énoncé donne : $P(D_{n+1} \mid M_n) = \frac{2}{10}$ et $P(D_{n+1} \mid D_n) = \frac{7}{10}$ et donc
$$d_{n+1} = 0,7d_n + 0,2m_n$$
3. On cherche à calculer $m_{n+1} = P(M_{n+1})$ On applique la formule des probabilités totales avec le SCE (M_N, D_N)
$$\begin{aligned} m_{n+1} &= P(M_{n+1} \mid M_n)P(M_n) + P(M_{n+1} \mid D_n)P(D_n) \\ &= P(M_{n+1} \mid M_n)m_n + P(M_{n+1} \mid D_n)d_n \end{aligned}$$
L'énoncé donne : $P(M_{n+1} \mid M_n) = \frac{8}{10}$ et $P(M_{n+1} \mid D_n) = \frac{3}{10}$ et donc
$$m_{n+1} = 0,3d_n + 0,8m_n$$
4. On a d'après la question précédente :
$$\begin{pmatrix} d_{n+1} \\ m_{n+1} \end{pmatrix} = A\begin{pmatrix} d_n \\ m_n \end{pmatrix}$$
C'est-à-dire
$$\boxed{X_{n+1} = AX_n}$$

On montre par récurrence la propriété P(n) : $X_n = A^n X_0$

— Initialisation $P(0)$ est vraie en effet, $A^0 = I_2$ donc $A^0 X_0 = X_0$

— Heredite On suppose la propriété vraie à un certain rang $n \in \mathbb{N}$ et on souhaite prouver $P(n+1)$. On a $X_{n+1} = AX_n = A(A^n X_0)$ par hypothése de récurrence. Donc

$$X_{n+1} = A^{n+1} X_0$$

et donc $P(n+1)$ est vraie.

— Conclusion

$$\boxed{\text{Pour tout } n \in \mathbb{N},\ X_n = A^n X_0}$$

— On obtient donc

$$X_n = \frac{1}{5} \begin{pmatrix} 3\,(1/2)^n + 2 & -2\,(1/2)^n + 2 \\ -3\,(1/2)^n + 3 & 2\,(1/2)^n + 3 \end{pmatrix} \begin{pmatrix} d_0 \\ m_0 \end{pmatrix}$$

Or d'après l'énoncé $d_0 = 1$ et $m_0 = 0$ donc

$$\boxed{d_n = \tfrac{1}{5}(3\,(1/2)^n + 2)}$$

Exercice 6. On modélise une carte d'un jeu de 52 cartes par une liste de deux éléments le premier sera sa couleur (on dispose d'une liste `couleur=['Pique', 'Coeur', 'Carreau' , 'Trefle']` que l'on pourra utiliser dans les fonctions) et la deuxième valeur son numéro. Pour simplifier le valet sera 11, la dame 12 et le roi 13. Ainsi la dame de carreau sera modélisé par `['Carreau', 12]` et le 6 de coeur par `['Coeur', 6]`.

On modélise le jeu de carte par une liste contenant toutes les cartes.

Afin de répondre aux différentes questions, on pourra utiliser les fonctions des questions précédentes même si elles n'ont pas été codées.

1. Ecrire une fonction Python `paquet` qui retourne une liste correspondant à la modélisation décrite d'un jeu de cartes.
2. Remplir la ligne suivante afin que la liste `L` soit affectée à la même liste mais sans le terme d'indice `i`.

```
L=L[...:...] + L[...:...]
```

3. Ecrire une fonction Python `tirage_sans_remise` qui prend en argument un entier `n` qui retourne une liste de `n` cartes tirées aux hasard et sans remise du paquet. (On n'utilisera pas la fonction `.pop`)
4. Dans la question précédente quelle condition doit on imposer sur `n` ? Que faudrait il ajouter pour vérifier cette condition ?
5. Que fait la fonction suivante :

```
def mystere(n):
  J=tirage_sans_remise(n)
  C=[c[0] for c in J]
  return(C)
```

6. Ecrire une fonction Python `check_couleur` qui prend en argument un entier `n`, selectionne `n` cartes sans remises et retourne `True` si on obtient `n` cartes de la même couleur et `False` sinon.
7. On se propose d'écrire une fonction qui permet de vérifier si les cartes tirées forment une suite. Pour cela il faut trier la liste des valeurs des cartes. Compléter la fonction suivante qui prend en argument un entier `n` et retourne la liste des cartes triées dans l'ordre croissant

```
def tri_carte(n):
   J=tirage_sans_remise(n)
   T= [J[0]]  #on place la premiere carte dans une liste
   for i in .......... : # on regarde toutes les cartes de J
      val_carte= .... # on regarde la valeur de la carte
      k=..... # On initialise le compteur
      while  k<len(T) and val_carte<T[k][1] : #on compare avec
                                          #les cartes deja tries
        k=..... #on augmente le compteur
      T = T[... : ...] + [J[i]] +T[ ... : ...]  #on  place la carte
                                          #au bon endroit dans T
   return(T)
```

8. Ecrire une fonction Python `check_suite` qui prend en argument un entier `n`, selectionne `n` cartes sans remises et retourne `True` si on obtient `n` cartes dont les valeurs se suivent et `False` sinon.

Correction 6.

```
def paquet():
  P=[]
  for i in range(1,14):
    for c in couleur:
      P.append([c,i])
  return(P)
```

```
L=L[:i]+L[i+1:]
```

```
import random as rd
def tirage_sans_remise(n):
  M=[]
  J=paquet()
  for i in range(n):
    x=randint(0,len(J)-1)
    M.append(x)
    J=J[:x]+J[x+1:]
  return(M)
```

4. Nécessairement $n \leq 52$, donc il faudrait rajouter une condition du type `if n<=52` avant la boucle
5. Retourne les couleurs d'un tirage de n cartes.

```
def check_couleur(n):
  C=mystere(n)
  c0=C[0]
  for c in C:
    if c!=c0:
      return False
  return True
```

```
def tri_carte(n):
   J=tirage_sans_remise(n)
   T= [J[0]]  #on place la premiere carte dans une liste
   for i in range(len(J)) : # on regarde toutes les cartes de J
      val_carte= J[i][1] # on regarde la valeur de la carte
      k=0 # On initialise le compteur
      while val_carte<T[k][1]  and k<len(T): #on compare avec
                                          #les cartes deja tries
        k=k+1 #on augmente le compteur
      T = T[ : k] + J[i] +T[ k+1 : ]  #on  place la carte
                                          #au bon endroit dans T
   return(T)
```

```
def check_suite(n):
  L=tri_carte(n)
  for i in range(1,len(L)):
    if L[i-1]!= L[i]:
      return( False)
  return(True)
```