

Devoir Surveillé 03

Le samedi 25 novembre 2023
durée 3h00

Documents écrits, électroniques, calculatrices et téléphones portables interdits

La plus grande attention sera apportée à la qualité de la rédaction, syntaxe et orthographe comprise.

Numérotez les copies et questions et soulignez ou encadrez les résultats.

*Vous pouvez utiliser les résultats des questions en amont d'une question donnée en la référant clairement
Les calculs, même longs et fastidieux doivent figurer sur la copie*

*Si au cours de l'épreuve, vous repérez ce qui vous semble être une erreur d'énoncé, vous le signalerez sur
votre copie et poursuivrez la composition.*

Exercice I

Partie A

Programmation Python de primitives de tirage au sort

On suppose que la bibliothèque Python numpy a été importée en début de script sous l'alias np.

A.1. Ecrire, en vous servant uniquement de la fonction `numpy.random.rand` pour l'obtention de nombres aléatoires une fonction d'entête `Bernoulli(p = 0.5)` retournant une valeur simulée d'une v.a de BERNOULLI de paramètre de succès p (ayant pour valeur par défaut $p = \frac{1}{2}$).

A.2. Ecrire, en vous servant uniquement de la fonction `numpy.random.rand` une fonction d'entête `Uniforme(N)` retournant une valeur simulée d'une v.a uniforme sur l'intervalle d'entiers $V = \{0, \dots, N-1\}$.

A.3. Compléter, en vous servant uniquement de la fonction `numpy.random.rand` la fonction d'entête `X(XX, pi)` de sorte qu'elle retourne une valeur simulée d'une v.a X prenant ses valeurs dans un ensemble fini \mathcal{X} , décrit par la liste Python `XX` avec probabilités spécifiées par la fonction $\pi : \mathcal{X} \mapsto [0, 1]$, décrite par la fonction Python `pi`.

Cette fonction a pour entête `pi(x)` où x est supposé être un élément de `XX`. La fonction π vérifie

$$\sum_{x \in \mathcal{X}} \pi(x) = 1.$$

```
def X(XX, pi) :  
    u = .....  
    q = 0.0  
    for x in XX :  
        q += .....  
        if u <= q :  
            return x
```

Ces trois fonctions devront être utilisées dans l'écriture des fonctions de la suite du problème comportant un élément aléatoire. En d'autres termes, nous n'aurez plus à écrire `numpy.random.rand` dans les fonctions définies dans les questions suivantes.

Partie B

Des ensembles de grand cardinal

Soit un ensemble V de cardinal N dont les éléments (les « v ») sont indexés par les entiers $i \in \{0, \dots, N-1\}$, c'est à dire :

$$V = \{v_i, i \in \{0, \dots, N-1\}\}$$

avec

$$\forall i, j \in \{0, \dots, N-1\}, v_i = v_j \Leftrightarrow i = j$$

On peut chercher –par exemple en physique statistique– à tirer au sort, en suivant une loi imposée, une fonction $x : V \rightarrow \{-1, +1\}$. Une telle fonction peut représenter par exemple une configuration de dipôles magnétiques placés en chaque point de V et dont l'orientation peut prendre deux valeurs opposées. La loi impose donne les configurations les plus probables, celles que nous observerons la plupart du temps.

Dans ce cas l'ensemble $\mathcal{X} = \{x : V \rightarrow \{-1, 1\}\}$ est un ensemble fini, on dispose d'une application $\pi : \mathcal{X} \rightarrow [0, 1]$ vérifiant

$$\sum_{x \in \mathcal{X}} \pi(x) = 1$$

et on cherche à construire une variable aléatoire X (définie sur un certain espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$) à valeurs dans \mathcal{X} et vérifiant :

$$\forall x \in \mathcal{X}, \mathbb{P}(X = x) = \pi(x)$$

A cet effet,

1. dans un premier temps, on construit une chaîne de MARKOV $(Y_k)_{k \geq 0}$ qui va parcourir tout l'ensemble \mathcal{X} et telle que la loi de Y_k est la loi uniforme sur \mathcal{X} ;
2. On introduit ensuite (partie C) une modification de cette chaîne, une autre chaîne de MARKOV $(X_k)_{k \geq 0}$ qui va parcourir tout l'ensemble \mathcal{X} et telle que la loi de X_k , lorsque k est grand est très proche de la loi sur \mathcal{X} spécifiée par π ;

On montre dans les questions B.1, B.2 la nécessité de disposer de méthodes plus efficaces que la méthode naïve décrite en question A.3.

B.1. Donner $\#\mathcal{X}$, le cardinal de \mathcal{X} , c'est à dire le nombre d'applications de V dans $\{-1, +1\}$, en fonction de N , le cardinal de V .

B.2. Programmation Python. Méthode naïve.

On suppose les conventions suivantes et variables et fonctions déclarées en début de script pour être utilisables dans les fonctions à écrire :

- les éléments de V sont identifiés aux entiers de $\{0, \dots, N-1\}$, on représente leur ensemble V par :

$$V = \text{np.arange}(N)$$

qui est un vecteur (tableau Numpy de forme $V.\text{shape} = (N,)$), comportant les entiers consécutifs entre 0 et $N-1$;

- Un élément x de \mathcal{X} est représenté par x , un vecteur (tableau Numpy de forme $V.\text{shape} = (N,)$) tel que $x[v] = +1$ ou -1 , la valeur de $x(v)$;

B.2.a. Expliquer pourquoi la fonction `ToutesLesConfigs` du script suivant compose et retourne la liste de tous les éléments de \mathcal{X} . On pourra par exemple détailler le fonctionnement de l'algorithme lorsque $N = 3$.

```
def ToutesLesConfigs():
    suffixes_a_ajouter = [+1, -1]
    L = [[]] #Liste à retourner pour N=0, la liste vide est la seule config.
    for _ in range(N):
        L2 = []
        for x in L:
            for xv in suffixes_a_ajouter :
```

```

L2.append( x + [xv] )
L = L2
return [np.array(x) for x in L ]

```

B.2.b. On suppose $N = 100$ (ce qui arriverait si par exemple les « v » dans V étaient les cases d'un damier). Pourquoi la méthode exposée et programmée en A.3 est-elle vouée à l'échec ? Indication: Evaluer le nombre d'éléments de XX et comparer à la taille de votre mémoire

B.3. Description de la chaîne $(Y_k)_{k \geq 0}$.

On définit un graphe –au sens informatique de ce terme– noté Γ dont l'ensemble des sommets est \mathcal{X} et l'ensemble des arêtes est \mathcal{E} , défini comme suit. Deux éléments x et y de \mathcal{X} sont voisins (c'est à dire que l'arête (x, y) est dans \mathcal{E}) si il existe exactement un $v_* \in V$ tel que :

$$x(v_*) = -y(v_*) \text{ et } \forall v \in V, v \neq v_* \Rightarrow x(v) = y(v)$$

En d'autres termes, les deux applications $x, y : \mathcal{X} \rightarrow \{-1, +1\}$ sont voisines dans le graphe Γ si elles sont égales partout sauf en exactement un point $v_* \in V$.

Pour $x \in \mathcal{X}$,

— on note $\mathcal{E}(x)$ l'ensemble des arêtes comportant x pour extrémités, il s'agit de l'ensemble à N éléments :

$$\mathcal{E}(x) = \{(x, y) \in \mathcal{E}, y \in V\}.$$

— on note $\mathcal{V}(x)$ l'ensemble des voisins de x , il s'agit de l'ensemble à N éléments :

$$\mathcal{V}(x) = \{y \in V, (x, y) \in \mathcal{E}\} = \{y \in V, (x, y) \in \mathcal{E}(x)\}.$$

L'ensemble d'arêtes \mathcal{E} est tel que l'on peut relier par des arêtes deux sommets x et y quelconques par un « chemin » comportant au plus N arêtes.

La chaîne $(Y_k)_{k \geq 0}$ est construite en :

1. tirant au sort uniforme sur \mathcal{X} la valeur de Y_0 ;
2. Sachant que pour un certain entier k , $Y_k = x$, Y_{k+1} vaut un certain y tiré au sort uniforme parmi les N voisins (dans le graphe $\Gamma = (\mathcal{X}, \mathcal{E})$) de x . Une façon de faire ce tirage au sort est de tirer V_k au sort uniforme sur l'ensemble à N éléments V et de changer l'élément $x(V_k)$ en $y(V_k) = -x(V_k)$. Autrement dit :

$$\forall v \in V, Y_{k+1}(v) = \begin{cases} Y_k(v) & \text{si } v \neq V_k \\ -Y_k(v) & \text{si } v = V_k \end{cases}$$

B.3.a. Implémenter une fonction d'entête $Y_0()$ retournant un élément x de \mathcal{X} tiré uniformément dans \mathcal{X} . On pourra choisir au fût et à mesure les valeurs ± 1 des $x(v)$ en tirant au sort indépendamment et uniformément ces valeurs.

B.3.b. Implémenter une fonction d'entête $UnVoisin(x)$ qui, partant de $Y_k = x$ retourne une valeur aléatoire y pour Y_{k+1} tirée uniformément parmi les configurations voisines de x .

B.3.c. Implémenter une fonction d'entête $YK(k)$ qui retourne une valeur simulée –suivant la méthode décrite– de Y_k .

B.4. Un peu de théorie.

B.4.a. Montrer que pour tout entier $k \in \mathbb{N}$,

$$\forall x, y \in \mathcal{X}, \mathbb{P}(Y_{k+1} = y | Y_k = x) = K(y, x)$$

où on a posé

$$K(y, x) = \begin{cases} 0 & \text{si } y \notin \mathcal{V}(x) \\ \frac{1}{N} & \text{si } y \in \mathcal{V}(x) \end{cases}$$

B.4.b. Montrer (par récurrence sur k) que pour tout entier $k \in \mathbb{N}$, Y_k est uniformément distribuée sur \mathcal{X} .

Partie C

Présentation de l'algorithme de METROPOLIS-HASTINGS.

On se donne, comme en partie B, l'ensemble \mathcal{X} (de cardinal 2^N) des applications $x : V \rightarrow \{-1, +1\}$. On définit la fonction $\pi : \mathcal{X} \rightarrow [0, 1]$ (la distribution voulue) en se donnant une fonction $Z : \mathcal{X} \rightarrow]0, +\infty[$, qui nous permet, en posant $Z_* = \sum_{x \in \mathcal{X}} Z(x)$, de définir

$$\forall x \in \mathcal{X}, \pi(x) = \frac{Z(x)}{Z_*}$$

Cette fonction π vérifie bien :

1. $\forall x \in \mathcal{X}, \pi(x) \in [0, 1]$;
2. $\sum_{x \in \mathcal{X}} \pi(x) = 1$;

Notre but est de définir, avec une définition implémentable en machine, une variable aléatoire X , à valeurs dans \mathcal{X} vérifiant

$$\forall x \in \mathcal{X}, \mathbb{P}(X = x) = \pi(x)$$

Nous allons définir une chaîne de MARKOV $(X_k)_{k \geq 0}$ sur \mathcal{X} de sorte que pour k suffisamment grand, X_k remplit, au moins approximativement, le rôle de X .

C.1. On modifie la construction de la chaîne de MARKOV $(Y_k)_{k \geq 0}$ pour construire la chaîne $(X_k)_{k \geq 0}$ en imposant les règles suivantes :

- on pose comme configuration de départ X_0 tirée au hasard, par exemple uniformément $X_0 = Y_0$ ou par une autre méthode ;
- sachant qu'à l'instant k , $X_k = x$, on construit la configuration X_{k+1} en
 - tirant au sort y à partir de x comme dans la construction de Y_{k+1} à partir de Y_k ;
 - tirant au sort une variable de BERNOULLI B_k de paramètre de succès

$$p_{y,x} = \min\left(1, \frac{Z(y)}{Z(x)}\right)$$

et en posant :

- si $B_k = 1$ (succès), $X_{k+1} = y$;
- si $B_k = 0$ (échec), $X_{k+1} = x$;

C.1.a. Implémenter une fonction d'entête `UnPasX(x)` qui, partant de $X_k = x$ retourne une valeur aléatoire y pour X_{k+1} simulée suivant la règle précédemment décrite. On suppose que la fonction $Z : \mathcal{X} \rightarrow]0, +\infty[$ est représentée en Python par une fonction d'entête `Z(x)` où `x` représente un élément $x \in \mathcal{X}$, retournant un flottant > 0 .

C.1.b. Implémenter une fonction d'entête `XK(k)` retournant une valeur simulée de X_k pour un k entier passé en argument.

On peut démontrer, mais ce n'est pas notre objet ici, que lorsque k est suffisamment grand, la loi de X_k est proche, en un sens quantifiable, de la loi $(\pi(x))_{x \in \mathcal{X}}$ comme attendu.

La première étape est de montrer que si on impose que X_0 suit la loi décrite par π alors tous les X_k suivent cette loi.

Exercice II

On définit la fonction $h : \mathbb{R} \rightarrow \mathbb{R}$ par la formule

$$\forall x \in \mathbb{R}, h(x) = \begin{cases} x \cdot \ln(x) & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$$

1. Questions préliminaires.

1.a. Démontrer que la fonction h est continue sur tout \mathbb{R} .

1.b. Déterminer le tableau de variations de h .

1.c. Tracer (soigneusement !) le graphe \mathcal{H} de h dans un repère orthonormé (O, \vec{i}, \vec{j}) .

On tracera sur le même graphe la droite \mathcal{D} d'équation $y = x$ en prenant soin de respecter les positions respectives de \mathcal{D} et \mathcal{H} . On demande de représenter essentiellement la situation pour la plage d'abscisses variant entre -1 et $e^2 \leq 7,4$ où $e \simeq 2,71828$ vérifie $\ln(e) = 1$.

Si X est une variable aléatoire réelle (définie sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$) de densité f , on dit que X admet une entropie si l'intégrale généralisée

$$\int_{-\infty}^{+\infty} h \circ f(t) dt$$

converge absolument et dans ce cas, on désigne par $H(X)$, l'entropie de X , la valeur de cette intégrale généralisée :

$$H(X) = - \int_{-\infty}^{+\infty} h \circ f(t) dt.$$

2. Un problème de bonne définition.

Soit X une variable aléatoire réelle. On suppose qu'il existe un intervalle ouvert de \mathbb{R} (possiblement de longueur infinie) I tel que

$$\forall x \in I, f(x) > 0 \text{ et } \forall x \notin I, f(x) = 0. \quad (*)$$

2.a. Montrer que $\mathbb{P}(X \in I) = 1$.

De ce fait, on peut, pour faire les calculs, supposer que X est à valeurs dans l'intervalle I et donc que $f(X)$ est à valeurs dans $]0, +\infty[$.

2.b. Montrer que X admet une entropie si et seulement si la variable aléatoire $Y = -\ln f(X)$ admet une espérance. Montrer que dans ce cas, on a :

$$H(X) = \mathbb{E}(-\ln f(X))$$

3. Soit U une variable aléatoire qui suit la loi uniforme sur $[a, b]$ où a et b sont des réels tels que $a < b$.

3.a. Démontrer que U admet une entropie. 3.b. Déterminer $H(U)$.

4. Soit T une variable aléatoire qui suit une loi exponentielle de paramètre $\lambda > 0$, de densité f .

4.a. Rappeler la valeur de l'espérance de T . 4.b. Démontrer que T admet une entropie $H(T)$ à déterminer.

5. Soit Z une variable aléatoire réelle de loi normale $\mathcal{N}(m, \sigma^2)$ où $m \in \mathbb{R}$ et $\sigma > 0$. On note ϕ la densité usuelle de la variable aléatoire Z .

5.a. Rappeler espérance et variance de Z , formule pour ϕ .

5.b. Calculer, pour $z \in \mathbb{R}$, $\ln \phi(z)$.

5.c. Montrer que Z admet une entropie et que

$$H(Z) = \frac{1 + \ln(2\pi\sigma^2)}{2}$$

6. Soit X une variable aléatoire réelle de densité f donnée (satisfaisant la propriété (*) présentée à la question 2 si cela peut simplifier la représentation des calculs). On suppose que X admet une entropie.

Soit a, b deux nombres réels avec $a > 0$.

6.a. Donner une densité de $f_{a,b}$ de $X' = a.X + b$.

6.b. Montrer que X' admet une entropie et exprimer celle-ci en fonction de celle de X .

7. Soit X une variable aléatoire réelle de densité f donnée et satisfaisant la propriété (*) présentée à la question 2 et Z une variable aléatoire réelle de loi normale $\mathcal{N}(m, \sigma^2)$ où $m \in \mathbb{R}$ et $\sigma > 0$. On note ϕ sa densité comme en 5.

On suppose que X admet $\sigma^2 > 0$ pour variance, m pour espérance et une entropie $H(X)$.

7.a. Evaluer $\int_{-\infty}^{+\infty} f(t) \cdot \ln \phi(t) dt$.

7.b. En déduire que $H(Z) - H(X) = \int_I f(t) \ln \frac{f(t)}{\phi(t)} dt$.

7.c. Démontrer l'inégalité :

$$\forall q \in]0, +\infty[, \ln q \geq 1 - \frac{1}{q}$$

7.d. En déduire que $H(Z) - H(X) \geq 0$.

On vient de démontrer que parmi les variables aléatoires X à densité d'espérance m et de variance $\sigma^2 > 0$, les variables gaussiennes (ou normales) ont l'entropie maximale.

Exercice III

On considère la fonction f définie sur $[0, 1]$ par :

$$\forall x \in [0, 1], f(x) = (1-x)e^{x+\frac{x^2}{2}}.$$

1.a. Démontrer que f réalise une bijection strictement décroissante de $[0, 1]$ dans lui-même.

1.b. En déduire que :

$$\forall x \in [0, 1], (1-x)e^x \leq e^{-\frac{x^2}{2}}.$$

2. On fixe un réel α appartenant à l'intervalle $] \frac{1}{3}, \frac{1}{2} [$ et on considère les deux suites $(x_n)_{n \in \mathbb{N}^*}$ et $(y_n)_{n \in \mathbb{N}^*}$ ci-dessous définies par :

$$\forall n \in \mathbb{N}^*, x_n = n^{-\alpha} \text{ et } y_n = f(x_n).$$

2.a. Justifier que :

$$\forall x \in [0, x_n], y_n e^{-\frac{x^2}{2}} \leq (1-x)e^x.$$

2.b. Démontrer que :

$$\lim_{n \rightarrow +\infty} x_n \sqrt{n} = +\infty.$$

2.c. Rappeler le développement limité de $\ln(1-x)$ en 0 à l'ordre 3 et en déduire que :

$$\lim_{n \rightarrow +\infty} y_n^n = 1.$$

3. On pose :

$$I_n = \int_0^1 ((1-x)e^x)^n dx.$$

3.a. Démontrer que :

$$y_n^n \int_0^{x_n} e^{-\frac{nx^2}{2}} dx \leq I_n \leq \int_0^1 e^{-\frac{nx^2}{2}} dx.$$

3.b. En utilisant un changement de variable simple, en déduire :

$$y_n^n \int_0^{x_n \sqrt{n}} e^{-\frac{x^2}{2}} dx \leq I_n \sqrt{n} \leq \int_0^{\sqrt{n}} e^{-\frac{x^2}{2}} dx.$$

3.c. En utilisant une variable aléatoire suivant la loi normale $\mathcal{N}(0, 1)$, déduire de ce qui précède que :

$$I_n \sqrt{n} \xrightarrow[n \rightarrow +\infty]{} \sqrt{\frac{\pi}{2}}.$$

On pose :

$$\forall n \in \mathbb{N}^*, \gamma_n = \int_0^{+\infty} t^n e^{-t} dt \text{ et } \sigma_n = e^{-n} \sum_{k=0}^n \frac{n^k}{k!}.$$

4.a. En utilisant une variable aléatoire suivant la loi exponentielle $\mathcal{E}(1)$, calculer γ_1 et à l'aide d'une intégration par parties, établir que $(\gamma_n)_{n \in \mathbb{N}}$ existe et vérifie la relation de récurrence :

$$\forall n \in \mathbb{N}^*, \gamma_{n+1} = (n+1)\gamma_n.$$

4.b. En déduire une expression simple de γ_n en fonction de n (cette expression simple sera naturellement prolongée en $n=0$ par $\gamma_0 = \int_0^{+\infty} e^{-t} dt = 1$).

5.a. Démontrer que :

$$\forall n \in \mathbb{N}^*, \sigma_n = \frac{e^{-n}}{n!} \sum_{k=0}^n \binom{n}{k} n^k \gamma_{n-k}.$$

5.b. En déduire que :

$$\forall n \in \mathbb{N}^*, \sigma_n = \frac{e^{-n}}{n!} \int_0^{+\infty} (n+t)^n e^{-t} dt.$$

5.c. En utilisant le changement de variable $u = n+t$, en déduire :

$$\forall n \in \mathbb{N}^*, \sigma_n = 1 - \frac{1}{n!} \int_0^n u^n e^{-u} du.$$

5.d. En utilisant enfin le changement de variable $x = 1 - \frac{u}{n}$, en déduire que :

$$\forall n \in \mathbb{N}^*, \sigma_n = 1 - \frac{n^{n+1}}{n! e^n} I_n.$$

6. On admet l'équivalent ci-dessous appelé formule de STIRLING :

$$n! \underset{n \rightarrow +\infty}{\sim} \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

En déduire que $(\sigma_n)_{n \in \mathbb{N}^*}$ converge et préciser sa limite.

PYTHON

AGRO-VETO

2020

Listes

`[]` ----- Créer une liste vide
`[a]*n` ----- Créer une liste avec n fois l'élément `a`
`L.append(a)` Ajoute l'élément `a` à la fin de la liste `L`
`L1 + L2` --- Concatène les deux listes `L1` et `L2`
`len(L)` ----- Renvoie le nombre d'éléments de la liste `L`
`L.pop(k)` --- Renvoie l'élément d'indice k de `L` et l'enlève de `L`
`L.remove(a)` Enlève une fois la valeur `a` de la liste `L`
`max(L)` ----- Renvoie le plus grand élément de la liste `L`
`min(L)` ----- Renvoie le plus petit élément de la liste `L`
`sum(L)` ----- Renvoie la somme de tous les éléments de la liste `L`

Numpy

`import numpy as np`
`np.array()` ----- Transforme une liste en matrice `numpy`
`np.linspace(a,b,n)` ----- Crée une matrice ligne de n valeurs uniformément réparties entre a et b (inclus)
`np.zeros([n,m])` ----- Crée la matrice nulle de taille $n \times m$
`np.eye(n)` ----- Crée la matrice identité de taille n
`np.diag(L)` ----- Crée la matrice diagonale dont les termes diagonaux sont les éléments de la liste `L`
`np.transpose(M)` ----- Renvoie la transposée de `M`
`np.dot(M,P)` ----- Renvoie le produit matriciel `MP`
`np.sum(W)` ----- Renvoie la somme de tous les éléments de `M`
`np.prod(W)` ----- Renvoie le produit de tous les éléments de `M`
`np.max(W)` ----- Renvoie le plus grand élément de `M`
`np.min(W)` ----- Renvoie le plus petit élément de `M`
`np.shape(M)` ----- Renvoie dans un couple le format de la matrice `M`
`np.size(W)` ----- Renvoie le nombre d'éléments de `M`

Logique

`a == b` ----- Teste l'égalité « $a = b$ »
`a != b` ----- Teste « $a \neq b$ »
`a < b` ----- Teste « $a < b$ »
`a <= b` ----- Teste « $a \leq b$ »
`a > b` ----- Teste « $a > b$ »
`a >= b` ----- Teste « $a \geq b$ »
`not A` ----- Renvoie la négation de `A`
`A and B` --- Renvoie « `A` et `B` »
`A or B` --- Renvoie « `A` ou `B` »
`True` ----- Constante booléenne « Vrai »
`False` ----- Constante booléenne « Faux »

Numpy.linalg

`import numpy.linalg as la`
`la.inv(M)` ----- Renvoie l'inverse de la matrice `M` si elle est inversible
`la.eigvals(M)` ----- Renvoie la liste des valeurs propres de `M`
`la.eig(M)` ----- Renvoie un couple `L,P` où `L` est la liste des valeurs propres de `M` et `P` la matrice de passage associée
`la.matrix_rank(M)` ----- Renvoie le rang de `M`

Random

`import numpy.random as rd`
`rd.rand()` ----- Simule une réalisation d'une variable $X \mapsto \mathcal{U}(0,1)$
`rd.randint(a,b)` --- Simule une réalisation d'une variable $X \mapsto \mathcal{U}([a,b[)$
`rd.gauss(0,1)` ----- Simule une réalisation d'une variable $X \mapsto \mathcal{N}(0,1)$
`rd.choice(l)` ----- Choisit aléatoirement un élément de la liste `L`

Math

`import numpy as np`
`np.atan(x)` ----- Renvoie `arctan(x)` `np.sqrt(x)` --- Renvoie \sqrt{x} si $x \geq 0$
`np.floor(x)` ----- Renvoie $\lfloor x \rfloor$ `np.log(x)` --- Renvoie $\ln(x)$ si $x > 0$
`np.factorial(n)` --- Renvoie $n!$ si $n \in \mathbb{N}$ `np.exp(x)` --- Renvoie e^x

Matplotlib.pyplot

`import matplotlib.pyplot as plt`
`plt.plot(X,Y,'+-r')` ----- Génère la courbe des points définis par les listes `X` et `Y` (abscisses et ordonnées) avec les options :

- symbole : `'.'` point, `'o'` rond, `'h'` hexagone, `'+'` plus, `'x'` croix, `'*'` étoile, ...
- ligne : `'-'` trait plein, `'--'` pointillé, `'.'` alterné, ...
- couleur : `'b'` bleu, `'r'` rouge, `'g'` vert, `'c'` cyan, `'m'` magenta, `'k'` noir, ...

`plt.bar(X,Y)` ----- Génère l'histogramme des points définis par les listes `X` et `Y` (abscisses et ordonnées)
`plt.axis('equal')` ----- Rend le repère orthornormé
`plt.xlim(xmin,xmax)` ----- Fixe les bornes de l'axe des abscisses
`plt.ylim(ymin,ymax)` ----- Fixe les bornes de l'axe des ordonnées
`plt.show()` ----- Affiche le graphique

Correction DS 03

Correction Ex.-1

Partie A

Programmation Python de primitives de tirage au sort

A.1.

```
def Bernoulli(p= 0.5):  
    return int(np.random.rand() < p)
```

A.2.

```
def Uniforme(N) :  
    return int ( np.floor(np.random.rand()*N))
```

A.3.

```
def X(XX,pi) :  
    u = np.random.rand()  
    q = 0.0  
    for x in XX :  
        q += pi(x)  
        if u <= q :  
            return x
```

Partie B

Des ensembles de grand cardinal

B.1. On a $\mathcal{X} = \{-1, +1\}^V$ (notation standard pour l'ensemble des fonctions $V \rightarrow \{-1, +1\}$) et $\#\mathcal{X} = 2^N$ car $\#\{-1, +1\} = 2$ et $\#V = N$ (cours).

B.2. Programmation Python. Méthode naïve.

B.2.a. Pour expliquer le fonctionnement de la fonction `ToutesLesConfigs`, observons l'exemple où $N = 3$ en décrivant le contenu des listes `L` et `L2` à chaque tour de boucle : 3 tours pour la boucle extérieure indexée par `_` (coquetterie Python permettant de ne pas vraiment nommer la variable de boucle) puis un certain nombre de tours pour les deux boucles imbriquées.

Tour 0 : `L` est une liste contenant la liste vide et dans la première boucle imbriquée `x` varie sur les éléments de `L`, il y a donc un tour. Pendant ce tour `x = []`, et la boucle la plus interne tourne sur deux valeurs :

`xv = +1` : `L2 = [[+1]]`

`xv = -1` : `L2 = [[+1] , [-1]]`

Tour 1 : `L` (dernière valeur connue de `L2`) est une liste contenant deux listes et dans la première boucle imbriquée `x` varie sur les éléments de `L` :

`x = [+1]` la boucle la plus interne tourne sur deux valeurs :

`xv = +1` : `L2 = [[+1, +1]]`

`xv = -1` : `L2 = [[+1, +1] , [+1, -1]]`

`x = [-1]` la boucle la plus interne tourne sur deux valeurs :

xv = +1 :L2 = [[+1,+1] , [+1,-1] , [-1,+1]]
 xv = -1 :L2 = [[+1,+1] , [+1,-1] , [-1,+1] , [-1,-1]]

Tour 2 : L (dernière valeur connue de L2) est une liste contenant quatre = 2^2 listes et dans la première boucle imbriquée x varie sur les éléments de L :

x = [+1,+1] la boucle la plus interne tourne sur deux valeurs :

xv = +1 :L2 = [[+1,+1,+1]]
 xv = -1 :L2 = [[+1,+1,+1] , [+1,+1,-1]]

x = [+1,-1] la boucle la plus interne tourne sur deux valeurs :

xv = +1 :L2 = [[+1,+1,+1] , [+1,+1,-1] , [+1,-1,+1]]
 xv = -1 :L2 = [.. , .. , .. , [+1,-1,-1]]

x = [-1,+1] la boucle la plus interne tourne sur deux valeurs :

xv = +1 :L2 = [.. , .. , .. , [+1,-1,-1] , [-1,+1,+1]]
 xv = -1 :L2 = [.. , .. , .. , .. , .. , [-1,+1,-1]]

x = [-1,-1] la boucle la plus interne tourne sur deux valeurs :

xv = +1 :L2 = [.. , .. , .. , .. , .. , [-1,+1,-1] , [-1,-1,+1]]
 xv = -1 :L2 = [.. , .. , .. , .. , .. , .. , [-1,-1,+1] , [-1,-1,-1]]

A la fin, la liste L (dernière valeur connue de L2) contient les huit = 2^3 listes décrivant les éléments de \mathcal{X} pour $N = 3$ dans ce cas.

B.2.b. On suppose $N = 100$. La liste XX qui doit être composée entièrement à l'avance contient $2^{100} = (1024)^{10} \geq 10^{30}$, c'est à dire contient mille milliards de milliards de milliards d'éléments. Aucune machine actuelle ne dispose d'une telle place en mémoire (un Go c'est un milliard d'octets, Un To mille milliards d'octets). D'un autre point de vue, sans même stocker les éléments un par un le nombre de tours de boucles a effectuer dans la fonction ToutesLesConfigs serait probablement du même ordre de grandeur, (une machine de bureau, c'est au grand maximum 1 Milliard de tours par seconde et il y a un milliard de secondes dans une trentaine d'années.)

B.3. Description de la chaîne $(Y_k)_{k \geq 0}$.

B.3.a.

```
def Y0() :
    x = np.zeros((N,))
    for i in V :
        x[i] = 2*Bernoulli() - 1 #tirage uniforme sur +/- 1
        #alternative x[i] = Uniforme(2)-1
    return x
```

B.3.b.

```
def UnVoisin(x) :
    y = x[:] #copie de x, pas obligatoire
    v_k = Uniforme(N) #v_k: là où on fait différer les deux x et y
    y[v_k] = - x[v_k]
    return y
```

B.3.c.

```
def YK(k) :
    x = Y0()
    for _ in range(k) :
        x = UnVoisin(x)
    return x
```

B.4. Un peu de théorie.

B.4.a. Soit $k \in \mathbb{N}$, $x, y \in V$.

— Si y n'est pas voisin de x, c'est à dire si $y \notin \mathcal{V}(x)$, sachant qu'à l'instant k , $Y_k = x$, la règle interdit

de sauter en y et donc $Y_{k+1} = 0$ est impossible. On a donc

$$\mathbb{P}(Y_{k+1} = y | Y_k = x) = 0$$

- Si y est voisin de x , c'est à dire si $y \in \mathcal{V}(x)$, sachant qu'à l'instant k , $Y_k = x$, la règle permet de sauter y avec probabilité (de transition) $\frac{1}{N}$ car x possède N voisins et on doit en choisir un uniformément. On a donc

$$\mathbb{P}(Y_{k+1} = y | Y_k = x) = \frac{1}{N}$$

Finalement,

$$\forall x, y \in V, \mathbb{P}(Y_{k+1} = y | Y_k = x) = \begin{cases} 0 & \text{si } y \notin \mathcal{V}(x) \\ \frac{1}{N} & \text{si } y \in \mathcal{V}(x) \end{cases} = K(y, x)$$

B.4.b. On montre que pour tout entier $k \in \mathbb{N}$, $H_k : Y_k$ est uniformément distribuée sur \mathcal{X} .

Init. Par choix de Y_0 , Y_0 est uniformément distribuée sur \mathcal{X} .

Hérédité. Supposons que pour un certain entier naturel k , Y_k est uniformément distribuée sur \mathcal{X} .

Calculons, pour $y \in \mathcal{X}$ donné, $\mathbb{P}(Y_{k+1} = y)$ en conditionnant suivant les valeurs possibles de Y_k c'est à dire en utilisant la formule des probabilités totales avec le SCEI $\{\{Y_k = x\}, x \in \mathcal{X}\}$. On a :

$$\mathbb{P}(Y_{k+1} = y) = \sum_{x \in \mathcal{X}} \mathbb{P}(Y_{k+1} = y | Y_k = x) \cdot \mathbb{P}(Y_k = x) = \sum_{x \in \mathcal{X}} K(y, x) \cdot \mathbb{P}(Y_k = x)$$

$$[\text{valeur de } K(y, x)] = \sum_{x: y \in \mathcal{V}(x)} \frac{1}{N} \cdot \mathbb{P}(Y_k = x)$$

$$[Y_k \text{ uniforme sur } \mathcal{X}] = \sum_{x: y \in \mathcal{V}(x)} \frac{1}{N} \cdot \frac{1}{2^N}$$

$$[\#\mathcal{V}(x) = N] = \frac{1}{2^N}$$

Ceci montre que $\mathbb{P}(Y_{k+1} = y)$ ne dépend pas de $y \in \mathcal{X}$ et donc que Y_{k+1} est uniformément distribuée sur \mathcal{X} .

Partie C

Présentation de l'algorithme de METROPOLIS-HASTINGS.

C.1.a.

```
def UnPasX(x) :
    y = UnVoisin(x)
    p = Z(y)/Z(x) #p = min(1, Z(y)/Z(x)) inutile car..
    b = Bernoulli(p = p) #.. si p >= 1, Bernoulli retourne 1..
    if b :
        return y
    else :
        return x
```

C.1.b.

```
def XK(k) :
    x = Y0()
    for _ in range(k) :
        print(x)
        x = UnPasX(x)
    return x
```

Correction Ex.-2 On définit la fonction $h : \mathbb{R} \rightarrow \mathbb{R}$ par la formule

$$\forall x \in \mathbb{R}, h(x) = \begin{cases} x \cdot \ln(x) & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$$

1. Questions préliminaires.

1.a. Il est clair que h est définie sur tout \mathbb{R} , qu'en tant que constante 0, elle est continue sur $] -\infty, 0[$ et que par produit de fonctions continues $x \mapsto x$ et $x \mapsto \ln(x)$, h est continue sur $]0, +\infty[$. Il reste donc à observer le comportement à la jonction :

$$\begin{aligned} \lim_{x \rightarrow 0^+} h(x) &= \lim_{x \rightarrow 0^+} x \cdot \ln(x) = 0 \text{ par croissances comparées de } x \text{ et } \ln x \text{ en } 0 \\ \lim_{x \rightarrow 0^-} h(x) &= \lim_{x \rightarrow 0^-} 0 = 0. \end{aligned}$$

Comme $0 = h(0)$, h est continue en 0.

1.b. Il nous suffit d'étudier h sur $]0, +\infty[$, intervalle sur lequel cette fonction est \mathcal{C}^1 avec

$$\forall x \in]0, +\infty[, h'(x) = 1 + \ln(x)$$

- Pour $x > 0$, $h'(x) = 0 \Leftrightarrow x = e^{-1}$ et par continuité, h' est de signe constant sur chacun des intervalles $]0, e^{-1}[$ et $]e^{-1}, +\infty[$.
- $\forall x \in]0, e^{-1}[$, $h'(x) < 0$, du fait de la limite $-\infty$ en 0^+ de h' ,
- $\forall x \in]e^{-1}, +\infty[$, $h'(x) > 0$, du fait de la limite $+\infty$ en $+\infty$ de h' ,

On a donc le tableau de signe et le tableau de variations de h sur $]0, +\infty[$:

x	0	e^{-1}	$+\infty$
$h'(x)$	$-\infty$ -	0	+ $+\infty$
$h(x)$	0	\searrow $-e^{-1}$	\nearrow $+\infty$

1.c. cf. Fig. 1 en p.5.

2. Un problème de bonne définition.

Soit X une variable aléatoire réelle. On suppose qu'il existe un intervalle ouvert de \mathbb{R} (possiblement de longueur de longueur infinie) I tel que

$$\forall x \in I, f(x) > 0 \text{ et } \forall x \notin I, f(x) = 0. \tag{*}$$

2.a. On a par définition du fait que f est densité de X :

$$\mathbb{P}(X \in I) = \int_I f(x) dx.$$

Comme f est nulle sur les (s'ils existent) intervalles de \mathbb{R} à gauche et à droite de I , on peut changer les bornes et obtenir :

$$\mathbb{P}(X \in I) = \int_{-\infty}^{+\infty} f(x) dx.$$

Comme f est une densité de probabilité sur \mathbb{R} , on a donc

$$\mathbb{P}(X \in I) = \int_{-\infty}^{+\infty} f(x) dx = 1.$$

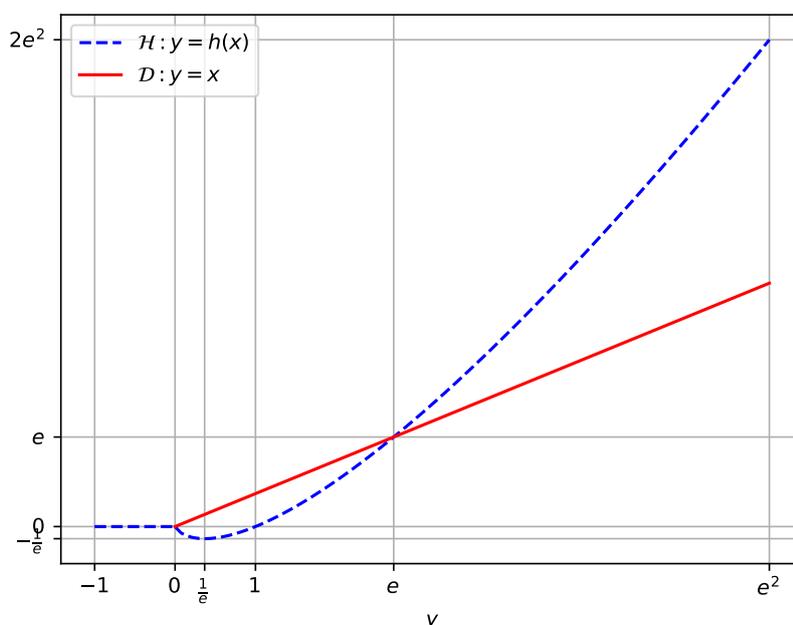


FIGURE 1 – Le graphe \mathcal{H} , la droite \mathcal{D} , le croisement est à l'abscisse $x = e$, l'intersection de \mathcal{H} avec l'axe des abscisses à $x = 1$.

De ce fait, on peut, pour faire les calculs, supposer que X est à valeurs dans l'intervalle I et donc que $f(X)$ est à valeurs dans $]0, +\infty[$.

2.b. Vu ce qu'on vient de dire, comme $h \circ f$ est nulle hors de l'intervalle I , l'IG $\int_{-\infty}^{+\infty} h \circ f(t) dt$ a même nature (et même valeur une fois la (A)CV acquise) que $\int_I h \circ f(t) dt = \int_I f(t) \cdot \ln f(t) dt$. Noter que cette IG est légitime car f est supposée continue, > 0 sur l'intervalle I et donc $t \mapsto f(t) \cdot \ln f(t)$ est bien définie, continue (bonne composition $\ln \circ f$ et produit) sur I et cette IG est légitime à considérer avec singularité(s) aux extrémités de I .

On a donc obtenu que X admet une entropie si et seulement si l'IG $\int_I f(t) \cdot \ln f(t) dt$ est ACV, en retour, ceci est équivalent (formule de transfert) au fait que $\ln f(X)$ admet une espérance et on a

$$\mathbb{E}(-\ln f(X)) = -\mathbb{E}(\ln f(X)) = -\int_I f(t) \cdot \ln f(t) dt = H(X).$$

3. Soit U une variable aléatoire qui suit la loi uniforme sur $[a, b]$ où a et b sont des réels tels que $a < b$.

3.a. La v.a. U , qui a pour densité $f : u \mapsto \frac{1}{b-a} \mathbb{1}_{\{u \in [a, b]\}}$ admet une entropie si et seulement si (car $\mathbb{1}_{\{U \in [a, b]\}} = 1$, la v.a constante égale à 1)

$$\ln \frac{1}{b-a} \mathbb{1}_{\{U \in [a, b]\}} = \ln \frac{1}{b-a}$$

admet une espérance. Dit comme ça, tout devient clair et **3.b.**

$$H(U) = -\ln \frac{1}{b-a}.$$

4. Soit T une variable aléatoire qui suit une loi exponentielle de paramètre $\lambda > 0$, de densité f .

4.a. $\mathbb{E}(T) = \frac{1}{\lambda}$. **4.b.** Comme T admet pour densité $f : t \mapsto \lambda \cdot e^{-\lambda \cdot t} \cdot \mathbb{1}_{\{t > 0\}}$, que l'on peut prendre $I =]0, +\infty[$, on peut supposer T à valeurs dans $]0, +\infty[$ et

$$f(T) = \lambda \cdot e^{-\lambda \cdot T}$$

On a donc $\ln f(T) = \ln \lambda - \lambda \cdot T$ et comme T admet une espérance, il vient que $\ln f(T)$ admet une espérance avec (linéarité de l'espérance)

$$\mathbb{E}(\ln f(T)) = \ln \lambda - \lambda \cdot \mathbb{E}(T) = \ln \lambda - 1$$

et donc

$$H(T) = 1 - \ln \lambda.$$

5. Soit Z une variable aléatoire réelle de loi normale $\mathcal{N}(m, \sigma^2)$ où $m \in \mathbb{R}$ et $\sigma > 0$. On note ϕ la densité usuelle de la variable aléatoire Z .

5.a. On a $\mathbb{E}(Z) = m$, $\mathbb{V}(Z) = \sigma^2$ et

$$\forall z \in \mathbb{R}, \phi(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{z-m}{\sigma}\right)^2}$$

5.b. On a donc, pour $z \in \mathbb{R}$,

$$\ln \phi(z) = -\frac{1}{2} \ln(2\pi) - \ln \sigma - \frac{1}{2} \left(\frac{z-m}{\sigma}\right)^2$$

5.c. On peut prendre $I = \mathbb{R}$ et on a donc

$$-\ln \phi(Z) = +\frac{1}{2} \ln(2\pi) + \ln \sigma + \frac{1}{2} \left(\frac{Z-m}{\sigma}\right)^2$$

Comme $\mathbb{E}((Z-m)^2) = \mathbb{V}(Z) = \sigma^2$, on a alors, par linéarité de l'espérance que $-\ln \phi(Z)$ admet une espérance (et donc que Z admet une entropie) avec

$$H(Z) = \frac{1}{2} \ln(2\pi) + \ln \sigma + \frac{1}{2} \frac{1}{\sigma^2} \mathbb{E}((Z-m)^2) = \frac{1 + \ln(2\pi\sigma^2)}{2}$$

6. Soit X une variable aléatoire réelle de densité f donnée et satisfaisant la propriété (*) présentée à la question 2. On suppose que X admet une entropie.

Soit a, b deux nombres réels avec $a > 0$.

6.a. On a vu dans le cours (et il faut ici *recopier cette démonstration*) que :

$$\forall x' \in \mathbb{R}, f_{a,b}(x') = \frac{1}{a} f\left(\frac{x'-b}{a}\right)$$

6.b. On a donc

$$\ln f_{a,b}(x') = -\ln a + \ln f\left(\frac{x'-b}{a}\right)$$

et vu que $X' = a \cdot X + b$, on a

$$\ln f\left(\frac{X'-b}{a}\right) = \ln f(X).$$

Finalement, par linéarité de l'espérance, X' admet une entropie avec

$$H(X') = -\mathbb{E}(\ln f_{a,b}(X')) = -\mathbb{E}(-\ln a + \ln f(X)) = +\ln a + H(X).$$

Noter que ce calcul est cohérent avec celui fait pour variables uniforme, exponentielle et normale où l'on peut à chaque fois se ramener à calculer l'entropie d'une variable normalisée de la même famille.

7. Soit X une variable aléatoire réelle de densité f donnée et satisfaisant la propriété (*) présentée à la question 2 et Z une variable aléatoire réelle de loi normale $\mathcal{N}(m, \sigma^2)$ où $m \in \mathbb{R}$ et $\sigma > 0$. On note ϕ sa densité comme en 5.

On suppose que X admet $\sigma^2 > 0$ pour variance, m pour espérance et une entropie $H(X)$.

7.a. On a vu que

$$\forall t \in \mathbb{R}, \ln \phi(t) = -\frac{1}{2} \ln(2\pi) - \ln \sigma - \frac{1}{2} \frac{1}{\sigma^2} (t - m)^2$$

et donc, par la formule de transfert (modulo les existence d'espérance qui s'avèrent, une fois écrites, avoir du sens car X admet une variance)

$$\begin{aligned} \int_{-\infty}^{+\infty} f(t) \cdot \ln \phi(t) dt &= \int_{-\infty}^{+\infty} f(t) \cdot \left(-\frac{1}{2} \ln(2\pi) - \ln \sigma - \frac{1}{2} \frac{1}{\sigma^2} (t - m)^2 \right) dt \\ &= -\frac{1}{2} \ln(2\pi) - \ln \sigma - \frac{1}{2} \frac{1}{\sigma^2} \underbrace{\mathbb{E}((X - m)^2)}_{=1} \\ &= -\frac{1}{2} \ln(2\pi) - \ln \sigma - \frac{1}{2} \\ \text{[cf. Q.5.c]} &= -H(Z) \end{aligned}$$

7.b. Et donc, par $\ln \frac{b}{a} = \ln(b) - \ln(a)$ lorsque $a, b > 0$,

$$\begin{aligned} H(Z) - H(X) &= \underbrace{\int_I f(t) \ln f(t) dt}_{-H(X)} + \left(\underbrace{- \int_I f(t) \ln \phi(t) dt}_{+H(Z)} \right) \\ \text{[IG ACV]} &= \int_I f(t) \ln \frac{f(t)}{\phi(t)} dt \end{aligned}$$

7.c. Posons pour $q > 0$,

$$g(q) = \ln q + \frac{1}{q}$$

La fonction g est \mathcal{C}^1 sur $]0, +\infty[$ avec

$$\forall q > 0, g'(q) = \frac{1}{q} - \frac{1}{q^2} = \frac{q-1}{q^2}$$

$$- \forall q > 0, g(q) = 0 \Leftrightarrow q = 1;$$

$$- \forall q > 0, g(q) < 0 \Leftrightarrow q < 1;$$

et donc g est strictement décroissante sur $]0, 1]$, strictement croissante sur $[1, +\infty[$ et donc

$$\forall q \in]0, +\infty[, g(q) \geq g(1) = 1, \text{ i.e. } \ln q \geq 1 - \frac{1}{q}$$

7.d. On peut maintenant minorer l'intégrande dans l'expression intégrale trouvée en 7.b (on utilise le fait que $\int_I \phi(t) dt \leq 1$) :

$$H(Z) - H(X) = \int_I f(t) \ln \frac{f(t)}{\phi(t)} dt \geq \int_I f(t) \left(1 - \frac{\phi(t)}{f(t)} \right) dt = \int_I f(t) dt - \int_I \phi(t) dt \geq 1 - 1 = 0.$$

On vient de démontrer que parmi les variables aléatoires X à densité d'espérance m et de variance $\sigma^2 > 0$, les variables gaussiennes (ou normales) ont l'entropie maximale.

Correction Ex.-3 On considère la fonction f définie sur $[0, 1]$ par :

$$\forall x \in [0, 1], f(x) = (1-x)e^{x+\frac{x^2}{2}}.$$

1.a. La fonction f est (par composition et produit de fonctions usuelles qui ne posent aucun problème de définition ni de régularité) de classe \mathcal{C}^1 sur $[0, 1]$ et on a

$$\forall x \in [0, 1], f'(x) = (1-x)(1+x)e^{x+\frac{x^2}{2}} - e^{x+\frac{x^2}{2}} = -x^2 \cdot e^{x+\frac{x^2}{2}}.$$

La fonction f' est donc < 0 sur $]0, 1[$ et, est donc strictement décroissante sur $[0, 1]$ (on peut fermer l'intervalle par continuité de f). On a $f(0) = 1$ et $f(1) = 0$ et donc (par le théorème de la bijection continue) f réalise une bijection (décroissante) de $[0, 1]$ dans lui-même.

1.b. On a montré que

$$\forall x \in [0, 1], (1-x)e^x \cdot e^{+\frac{x^2}{2}} \leq 1$$

et donc, du fait que $\forall X, \exp(-X) > 0$, en multipliant les termes de l'inégalité par $\exp(-X)$ avec $X = +\frac{x^2}{2}$, il vient que :

$$\forall x \in [0, 1], (1-x)e^x \leq e^{-\frac{x^2}{2}}.$$

2. On fixe un réel α appartenant à l'intervalle $]\frac{1}{3}, \frac{1}{2}[$ et on considère les deux suites ci-dessous définies par :

$$\forall n \in \mathbb{N}^*, x_n = n^{-\alpha} \text{ et } y_n = f(x_n).$$

2.a. Soit $x \in [0, x_n]$, on a donc, par décroissance de f sur $[0, x_n]$ que

$$f(x) \geq f(x_n) = y_n$$

C'est à dire :

$$(1-x)e^x e^{+\frac{x^2}{2}} \geq y_n$$

et, comme précédemment, il vient que :

$$\forall x \in [0, x_n], y_n e^{-\frac{x^2}{2}} \leq (1-x)e^x.$$

2.b. On a $x_n \sqrt{n} = n^{-\alpha+\frac{1}{2}}$. Comme $-\alpha + \frac{1}{2} > 0$, il vient que :

$$\lim_{n \rightarrow +\infty} x_n \sqrt{n} = +\infty.$$

2.c. On a, lorsque $x \rightarrow 0$,

$$\ln(1-x) = -x - \frac{1}{2}x^2 - \frac{1}{3}x^3 + o(x^3)$$

et, comme $y_n = (1-x_n)e^{x_n+\frac{1}{2}x_n^2}$, on a, vu que $x_n \xrightarrow{n \rightarrow +\infty} 0$,

$$y_n^n = \exp(n \cdot \ln(1-x_n)) = \exp\left(-n \cdot x_n - \frac{1}{2}n \cdot x_n^2 - \frac{1}{3}n \cdot x_n^3 + n \cdot o(x_n^3) + x_n \cdot n + \frac{1}{2}n \cdot x_n^2\right) = \exp\left(-\frac{1}{3}n \cdot x_n^3 + n \cdot o(x_n^3)\right)$$

Or $n \cdot x_n^3 = n^{1-3\alpha} \rightarrow 0$ car $1-3\alpha < 0$ et donc

$$\exp\left(-\frac{1}{3}n \cdot x_n^3 + n \cdot o(x_n^3)\right) \rightarrow 1$$

et donc

$$\lim_{n \rightarrow +\infty} y_n^n = 1.$$

3. On pose :

$$I_n = \int_0^1 ((1-x)e^x)^n dx.$$

3.a. En utilisant les inégalités de 2.a (pour l'inégalité de gauche) et de 1.b (pour l'inégalité de droite) que l'on porte à la puissance n (tous les termes sont positifs), il vient que

$$\forall x \in [0, 1], y_n^n e^{-\frac{nx^2}{2}} \leq ((1-x)e^x)^n \leq e^{-\frac{nx^2}{2}}$$

En intégrant cette famille d'inégalités sur leur segment de validité, par croissance de l'intégrale, il vient que :

$$y_n^n \int_0^{x_n} e^{-\frac{nx^2}{2}} dx \leq I_n \leq \int_0^1 e^{-\frac{nx^2}{2}} dx.$$

3.b. Posont $z = (\sqrt{n} \cdot x)$ dans chacune des deux intégrales, il vient (changement de variable linéaire, $dx = \frac{1}{\sqrt{n}} dz$, bornes multipliée par \sqrt{n}) :

$$y_n^n \int_0^{x_n \sqrt{n}} e^{-\frac{z^2}{2}} dx \leq I_n \sqrt{n} \leq \int_0^{\sqrt{n}} e^{-\frac{z^2}{2}} dx.$$

3.c. Comme $x_n \sqrt{n} \rightarrow +\infty$, $\sqrt{n} \rightarrow +\infty$, les deux intégrales de droite et gauche ont pour limite commune la demi-intégrale gaussienne $\int_0^{+\infty} e^{-\frac{1}{2}x^2} dx = \frac{1}{2}\sqrt{2\pi} = \sqrt{\frac{\pi}{2}}$, enfin, comme $y_n^n \rightarrow 1$, par le théorème des gendarmes :

$$I_n \sqrt{n} \xrightarrow[n \rightarrow +\infty]{} \sqrt{\frac{\pi}{2}}.$$

On pose :

$$\forall n \in \mathbb{N}^*, \gamma_n = \int_0^{+\infty} t^n e^{-t} dt \text{ et } \sigma_n = e^{-n} \sum_{k=0}^n \frac{n^k}{k!}.$$

4.a. On a $\gamma_1 = \mathbb{E}(T) = 1$ où $T \sim \mathcal{E}(1)$. L'IG donnant la valeur présumée de γ_n est convergente du fait d'une majoration (+TCIP+ACV \Rightarrow CV) du type $t^n e^{-t} = o_{t \rightarrow +\infty} \left(\frac{1}{t^2}\right)$ et donc la suite $(\gamma_n)_{n \in \mathbb{N}}$ à valeurs réelles-existe. Cette suite, à l'aide d'une intégration par parties que le lecteur est prié d'aller retrouver dans son cours pour la recopier, vérifie la relation de récurrence :

$$\forall n \in \mathbb{N}^*, \gamma_{n+1} = (n+1)\gamma_n.$$

4.b. Là encore (par une récurrence de vérification), on a

$$\forall n \in \mathbb{N}, \gamma_n = n!.$$

5.a. Soit $n \in \mathbb{N}^*$. On a, en exprimant $\binom{n}{k}$ et γ_{n-k} à l'aide factorielles :

$$\frac{e^{-n}}{n!} \sum_{k=0}^n \binom{n}{k} n^k \gamma_{n-k} = \frac{e^{-n}}{n!} \sum_{k=0}^n \frac{n!}{k!(n-k)!} n^k (n-k)! = e^{-n} \cdot \sum_{k=0}^n \frac{1}{k!} n^k = \sigma_n$$

ce qu'il fallait démontrer (CQFD)

5.b. En exprimant γ_{n-k} sous forme intégrale, on a, par ailleurs, pour $n \in \mathbb{N}^*$:

$$\sigma_n = \frac{e^{-n}}{n!} \sum_{k=0}^n \binom{n}{k} n^k \gamma_{n-k} = \frac{e^{-n}}{n!} \sum_{k=0}^n \binom{n}{k} n^k \int_0^{+\infty} t^{n-k} e^{-t} dt$$

et donc par linéarité de l'intégrale, binôme de NEWTON, regroupement des exponentielles :

$$\begin{aligned}\sigma_n &= \int_0^{+\infty} \frac{e^{-n}}{n!} \sum_{k=0}^n \binom{n}{k} n^k t^{n-k} e^{-t} dt \\ &= \int_0^{+\infty} \frac{1}{n!} (t+n)^n e^{-(t+n)} dt\end{aligned}$$

5.c. et finalement, par le changement de variable affine $u = t + n$, $du = dt$, bornes décalées de n

$$\sigma_n = \int_n^{+\infty} \frac{1}{n!} u^n e^{-u} du$$

Par CHASLES ($\int_0^{+\infty} - \int_0^n = \int_n^{+\infty}$),

$$\sigma_n = \frac{1}{n!} \gamma_n - \int_0^n \frac{1}{n!} u^n e^{-u} du = 1 - \frac{1}{n!} \int_0^n u^n e^{-u} du.$$

5.d. Enfin le changement de variable affine décroissant $x = 1 - \frac{u}{n}$, $dx = -\frac{1}{n} du$, ($u : 0 \nearrow n$)/($x : 1 \searrow 0$), on a :

$$\forall n \in \mathbb{N}^*, \sigma_n = 1 - \frac{1}{n!} \int_0^n u^n e^{-u} du = 1 - \frac{n^{n+1}}{n!} e^{-n} \int_0^1 ((1-x)e^x)^n dx = 1 - \frac{n^{n+1}}{n! e^n} I_n.$$

6. On admet l'équivalent ci-dessous appelé formule de STIRLING :

$$n! \underset{n \rightarrow +\infty}{\sim} \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

On en déduit que

$$\frac{n^{n+1}}{n! e^n} I_n \sim \frac{n}{\sqrt{2\pi n}} I_n = \frac{\sqrt{n}}{\sqrt{2\pi}} I_n \rightarrow \frac{\sqrt{\pi}}{\sqrt{2.2\pi}} = \frac{1}{2}$$

et donc $(\sigma_n)_{n \in \mathbb{N}^*}$ converge vers $1 - \frac{1}{2} = \frac{1}{2}$.