CLASSE DE 2TSI
PREPARATION PLANCHES ORAUX PYTHON

Planche 01. CCS 2018

1. Programmer une fonction récursive $BINOM(n, p)$ qui renvoie nk. On renverra 0 si $k < 0$ ou $k > n$.
2. Une variable aléatoire X suit une loi binomiale $\mathcal{B}(n, p)$. Écrire une fonction PYTHON $rep(t, n, p)$ qui renvoie $P(X \leqslant t)$. (On utilisera la fonction $BINOM$.)
3. Tracer la fonction $t \mapsto rep(t, n, p)$ pour $t \in [-1, n+1]$ pour $(n, p) = (5, 0.3)$ et $(n, p) = (15, 0.7)$.

Planche 02. CCS 2017

L est une liste de valeurs entières. Écrire les fonctions :

1. f_1 qui renvoie le nombre de valeurs négatives ou nulles de L, puis son nombre de valeurs positives, dans cet ordre.
2. f_2 qui renvoie la somme des cubes des valeurs négatives plus la somme des carrées des valeurs positives.
3. f_3 qui renvoie le maximum de L et les indices correspondant à cette valeur.

Planche 03. CCS 2019

On pose $\phi : \mathcal{M}_n(\mathbf{R}) \to \mathcal{M}_n(\mathbf{R})$, $M \mapsto \mathrm{Tr}\,(M)\, I_n$ et $\psi : \mathcal{M}_n(\mathbf{R}) \to \mathcal{M}_n(\mathbf{R})$, $M \mapsto M - \phi(M)$.

1. Écrire une fonction Python qui renvoie la trace d'une matrice.
2. Écrire des fonctions Python qui permettent de calculer $\phi(M)$ et $\psi(M)$.
3. Tester ces trois fonctions pour des matrices choisies aléatoirement et commenter.

Planche 04. CCS 2019

On considère $G : x \mapsto \displaystyle\int_{\frac{1}{x}}^{x} \frac{t\ \arctan t}{(1+t^2)^2}\, dt$

1. Écrire une fonction Python G, d'argument x, qui renvoie $G(x)$.
2. Tracer côte à côte le graphe de G avec la droite $y = \dfrac{\pi}{8}$ et le graphe de la fonction $y = \dfrac{\pi}{8}\dfrac{1-x^2}{1+x^2}$.

Planche 05. CCS 2019

Calculer $I_n = \displaystyle\int_0^{\pi} \ln\left(2\sin\frac{x}{2}\right)\cos(nx)\, dx$ par la méthode des trapèzes.
(On prendra $n = 1000$ et on choisira 0.001 comme borne.)

Planche 06. ENSAM 2019

On cherche des entiers naturels non nuls n tels que $n = 2^k 3^l 5^m$ avec $(k, l, m) \in \mathbb{N}^3$. On les appelle entiers de type H.

1. Écrire une fonction booléenne d'argument n renvoyant $TRUE$ si n est un entier de type H (méthode récursive ou itérative).
   Afficher les 100 premiersentiers de type H.
2. Si n est un entier de type H, que peut-on dire de $\dfrac{n}{3}$, de $\dfrac{n}{5}$, de $\dfrac{n}{2}$?

3. En déduire une fonction qui prend en argument une liste d'entiers de type H, triée par ordre croissant et qui retourne l'entier de type H suivant.

Planche 07. CCINP

On considère dans $\mathbf{R}^3$ euclidien, rapporté à $(O, \vec{i}, \vec{j}, \vec{k})$ orthonormé direct, la surface (S) d'équation cartésienne :

$$x^2 + y^2 + z^2 + 2xyz - 1 = 0.$$

Écrire en langage Python une fonction booléenne retournant $TRUE$ si le point $M(x, y, z)$ appartient à (S) et $FALSE$ sinon.
Tester avec le point $(0, 0, 0)$.

Planche 08. CCINP

Soit f la fonction 2π-périodique définie par : $\forall x \in [-\pi, \pi]$, $f(x) = \dfrac{\pi^2}{12} - \dfrac{x^2}{4}$.

1. Tracer avec le logiciel le graphe de f sur $[-2\pi, 2\pi]$. Que remarque t-on ?
2. Déterminer les coefficients de Fourier de f.
3. Montrer que pour tout $x \in [-\pi, \pi]$, $\dfrac{\pi^2}{12} - \dfrac{x^2}{4} = \displaystyle\sum_{n=1}^{+\infty} (-1)^{n+1} \frac{\cos(nx)}{n^2}$.

   En déduire la valeur de $\displaystyle\sum_{n=1}^{+\infty} \frac{(-1)^{n+1}}{n^2}$.
4. Illustrer à l'aide de Python le dernier résultat précédent en écrivant une fonction prenant en paramètre un entier N non nul et renvoyant la valeur de la somme partielle $\displaystyle\sum_{k=1}^{N} \frac{(-1)^{k+1}}{k^2}$.

   Calculer ces sommes partielles avec $N = 10$, $N = 100$ et $N = 1000$ puis comparer avec $\pi^2/12$.

Planche 09. CCINP

Le plan P est rapporté à $(O, \vec{i}, \vec{j})$ orthonormé direct.

On considère la courbe Γ de représentation paramétrique $\begin{cases} x(t) &= \dfrac{2t-1}{t^2-1} \\ y(t) &= \dfrac{t^2+3}{t-1} \end{cases}$.

1. Faire le tableau des variations de $t \mapsto x(t)$ et de $t \mapsto y(t)$.
2. Justifier l'existence d'asymptotes verticales et horizontales pour (Γ).
3. Définir en Python les fonctions *abscisse*(t) et *ordonnee*(t), renvoyant respectivement $x(t)$ et $y(t)$.
4. Tracer avec Python la courbe $t \mapsto \dfrac{y(t)}{x(t)}$ sur $I_1 = \left[\dfrac{2}{3}, 1\right[$. Faire de même sur $I_2 = \left]1, \dfrac{4}{3}\right]$.
5. Tracer la courbe $t \mapsto y(t) - 8x(t)$ sur I_1 et sur I_2. Que constate t-on ? En déduire une asymptote oblique à (Γ).
6. Faire le tracé à la main de (Γ). Illustrer avec Python en tracant l'arc (Γ) et en incluant dans le dessin les diverses asymptotes.

Planche 10. CCINP

Un message doit transiter par un certain nombre de personnes. Chaque personne est susceptible soit de transmettre le message fidèlement à la suivante, soit de mentir et de lui transmettre son exact contraire. La probabilité qu'une personne mente est notée $p \in]0, 1[$. On suppose que les personnes sont toutes indépendantes les unes des autres (pour le choix de leur réponse). On note p_n la probabilité que le message transmis par la $n^{\text{ème}}$ personne soit identique au message initial. On conviendra que $p_0 = 1$.

1. Proposer une fonction Python $Chaine(n, p)$ qui simule la réalisation de n transmissions, n étant un entier naturel non nul fourni en paramètre et p la probabilité définie dans l'énoncé. Elle renverra 1 si le message transmis par la $n^{\text{ème}}$ personne est identique au message d'origine et 0 sinon.
2. On répète N chaînes toutes de mêmes paramètres n et p. À l'aide de la fonction précédente, proposer une fonction $simul(N, n, p)$ qui renvoie la proportion de messages correctement transmis par la $n^{\text{ème}}$ personne au cours de N réalisations de cette même chaîne.
   Tester la fonction pour différentes valeurs de p. (On pourra prendre $n = 200$ et $N = 3000$). Que constate-t-on ?
3. À l'aide de la formule des probabilité totales, montrer que pour tout entier n, $p_{n+1} = p+(1-2p)p_n$. Montrer que $q_n = p_n - 1/2$ est une suite géométrique dont on donnera la raison. En déduire l'expression de p_n en fonction de n et trouver $\lim\limits_{n\to+\infty} p_n$. Commenter ce résultat.

Planche 11. CCINP

Soit f, de matrice $M = \dfrac{1}{9}\begin{pmatrix} 8 & 1 & -4 \\ -4 & 4 & -7 \\ 1 & 8 & 4 \end{pmatrix}$ dans la base canonique.

1. Écrire en langage Python une fonction $monProd(a, b)$ permettant de calculer le produit scalaire de deux vecteurs de $\mathbf{R}^3$ appelés a et b à l'aide d'une boucle. Comparer avec $numpy.vdot(a, b)$.
2. Utiliser $monProd$ pour montrer que les colonnes de M forment une famille orthonormale.
3. Retrouver ce résultat toujours avec Python par le calcul de $M^T M$.
4. Toujours avec Python, justifier que f est une rotation.

Planche 12. CCS 2019. PSI

On note $A_n \in \mathcal{M}_n(\mathbf{R})$ de coefficient $a_{i,j} = \max(i, j)$.

1. Écrire un programme Python qui renvoie A_n et la représenter pour $n \in \{3, 4, 5\}$.
2. A_n est-elle inversible ? Calculer A_n^{-1} avec Python pour $n \in \{3, 4, 5\}$.
3. A_n est-elle diagonalisable ? Donner la dimension de ses sous-espaces propres pour $n \in \{3, 4, 5\}$ puis conjecturer la dimension des sous-espaces propres dans le cas général et prouver cette conjecture.

Planche 13. CCS 2019. PSI

Soient $h > 0$, suffisamment petit, et la suite (t_i) définie pour tout $i \in \mathbf{N}$ par $t_i = ih$. L'équation différentielle linéaire (E) : $y'' + \omega^2 y = 0$ régit le comportement d'un oscillateur harmonique.

1. Montrer que l'ensemble des solutions de (E) est en bijection avec l'ensemble des solutions de (E') :
$$X' = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} X.$$
2. Justifier que les suites (y_i) et (z_i) définies par :
$$\begin{cases} y_{i+1} = y_i + hz_i \\ z_{i+1} = z_i - h\omega^2 y_i \end{cases}$$
   fournissent une approximation des solutions de (E') puis de (E).
3. Écrire une fonction $Euler$ qui affiche la solution obtenue par la méthode d'Euler de (E). On tracera aussi la solution exacte de (E). Que remarque t-on après plusieurs périodes ?

Planche 14. ENSAM 2018. PSI

On donne $a_0 = u$, $b_0 = v$, $a_{n+1} = \sqrt{a_n b_n}$ et $b_{n+1} = \dfrac{1}{\frac{1}{a_n} + \frac{1}{b_n}}$.

1. Définir une fonction $iterer(p)$ qui renvoie $[a_{n+1}, b_{n+1}]$ à partir de $p = [a_n, b_n]$. Tester cette fonction pour $p = [3, 2]$.
2. Définir une fonction $suite$ (non récursive) qui prend (a_0, b_0, n) en argument et qui renvoie a_n et b_n. La tester pour $(3, 2, 3)$ et pour $(3, 2, 5)$. Que peut-on dire ?
3. On admet que (a_n) et (b_n) sont adjacentes, convergentes vers L tel que pour tout $n \in \mathbf{N}$,
$$a_n \leqslant L \leqslant b_n.$$
Créer une fonction $moyenne(a_0, b_0)$ qui renvoie L à 10^{-10} près. Quel est le n correspondant ? Tester cette fonction pour $(3, 2)$.
4. Tracer $moyenne(1, x)$ pour x allant de 1 à 10 avec un pas de 0.1.
5. Écrire une fonction récursive $Recsuite$ qui renvoie a_n et b_n. Comparer la vitesse d'execution de $Recsuite(3, 2, 10)$ et de $suite(3, 2, 10)$.

Planche 15. ENSAM 2018. PSI

1. Que fait le programme suivant ?
$>>>$ *import numpy.random as rd*
$>>>$ $LR\ =\ rd.random(6)\,;\ LR\ <\ 0.5$
$>>>$ $1 * (LR\ <\ 0.5)$
2. Définir la fonction $tirer(n)$ réalisant n expériences de Bernoulli de paramètre $1/2$.
3. Écrire une fonction $temps(Seq)$ qui, à partir d'une séquence de 0 et de 1, renvoie le nombre de tirages nécessaires pour que la séquence Seq apparaisse. Par exemple, si $Seq = [1, 0, 1, 1]$, et si les tirages sont $[0, 1, 0, 0, 1, 0, 1, 0, 1, 1]$ alors la fonction renvoie 10.