L'École Normale Supérieure de l'Enseignement Technique de Mohammedia

Filières Ingénieure-

COMPÉTENCES NUMÉRIQUES ET INFORMATIQUE

— TD N°6: —

Calcul Formel, Probabilités & Représentation Graphique

(Librairies principales : SymPy, NumPy, SciPy, Matplotlib, Seaborn)

Exercice N° 1 : Calcul Formel (SymPy)

Cette section introduit la manipulation symbolique pour la résolution de problèmes analytiques simples.

1. Dérivée et Racine :
   Soit la fonction $f(x) = x^3 - 6x^2 + 5x + 12$.
   - Déterminez symboliquement la dérivée première $f'(x)$.
   - Déterminez les valeurs de x qui annulent $f'(x)$ (les points critiques).
2. Développement et Simplification :
   - Simplifiez l'expression trigonométrique : $\sin(x)\cos(y) + \cos(x)\sin(y)$.
   - Développez l'expression $(2x + 3)^4$ et affichez le polynôme résultant.
3. Intégrale Définie :
   Calculez l'intégrale définie $\int_0^{\frac{\pi}{2}} \sin(x)\cos(x)\,dx$ et affichez le résultat exact (fractionnel si possible).
4. Résolvez symboliquement l'équation linéaire simple : 4x + 7 = 3(x - 1).
5. Déterminez la limite de la fonction $h(x) = \frac{e^{2x} - 1}{x}$ lorsque x tend vers 0.
6. Système Linéaire Matriciel :
   Résolvez le système d'équations linéaires suivant en utilisant les outils matriciels de SymPy
   $$\begin{cases} 2x - 3y = 5 \\ -x + 2y = -3 \end{cases}$$
7. **DÉFI,** EDO du Premier Ordre :
   Résolvez symboliquement l'équation différentielle du premier ordre : $\frac{dy}{dx} - y = x^2$.

Cette solution est demandée sous sa forme générale (avec constante d'intégration).

Exercice N° 2 : Probabilités et Simulation (NumPy, SciPy.stats)

Cette section utilise la simulation et les lois de probabilité pour l'analyse stochastique

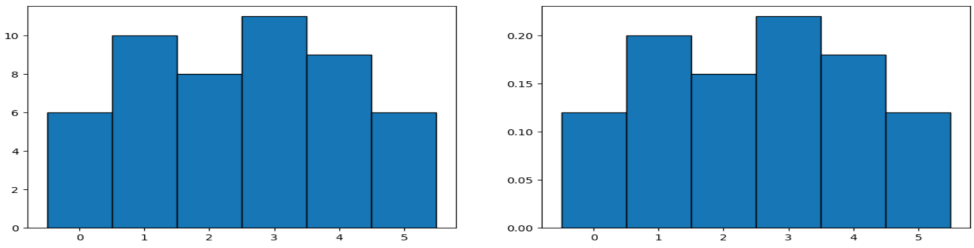
1. Cas N°1
   On lance 3 fois de façon indépendante une pièce donnant PILE avec la probabilité 13. On note X la variable aléatoire égale au nombre de PILE obtenus.
   - Que permet de simuler l'exécution du programme suivant ?

```
import numpy.random as rd

x=rd.random()
if x<1/3:
    piece="PILE"
else:
    piece="FACE"
print(piece)
```

- Écrire une fonction telle que l'exécution de simul_X() renvoie une réalisation de la variable aléatoire X.

L'exécution des lignes ci-dessous renvoie successivement la figure de gauche puis celle de droite.

```
import numpy.random as rd
import matplotlib.pyplot as plt

L=rd.randint(0,6,50)
Labs=[-0.5+k for k in range(0,7)]
plt.hist(L,Labs,edgecolor='k')
plt.show()
plt.hist(L,Labs,edgecolor='k',density=True)
plt.show()
```



- Réaliser 10000 simulations de la variable aléatoire X et représenter l'histogramme des données ainsi obtenues.

2. Cas N°2

Soit n un entier naturel non nul.
On effectue une série illimitée de tirages d'une boule avec remise dans une urne contenant n boules numérotées de 1 à n. Pour tout entier naturel k non nul, on note X_k la variable aléatoire égale au numéro de la boule obtenue au k-ième tirage.
Pour tout entier naturel k non nul, on note S_k la somme des numéros des boules obtenues lors des k premiers tirages :

$$S_k = \sum_{i=1}^{k} X_i.$$

On considère enfin la variable aléatoire T_n égale au nombre de tirages nécessaires pour que, pour la première fois, la somme des numéros des boules obtenues soit supérieure ou égale à n.

Exemple : avec n=10, si les numéros obtenus aux cinq premiers tirages sont dans cet ordre 2,4,1,5,9, alors on obtient : $S_1 = 2$, $S_2 = 6$, $S_3 = 7$, $S_4 = 12$, $S_5 = 21$ et $T_{10} = 4$.

On rappelle qu'en Python rd.randint(1,n+1) renvoie un entier au hasard et uniformément compris entre 1,n. Compléter la fonction du programme Python suivant, qui prend en argument le nombre n de boules contenues dans l'urne, pour qu'elle affiche une simulation de la variable aléatoire T_n.

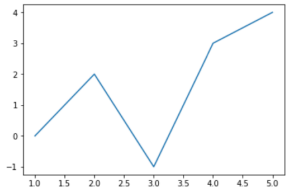
```
import numpy as np
import numpy.random as rd
def T(n) :
    S=..........
    T=..........
    while ..............:
        tirage=rd.randint(1,n+1)
        S=..................
        T=..................
    return T
```

Exercice N° 3 : Représentation Graphique Avancée (Matplotlib, Seaborn)

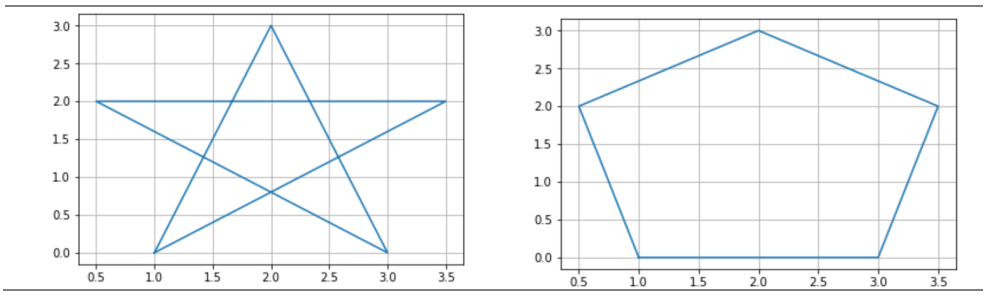
Cette section se concentre sur la visualisation de fonctions complexes et l'analyse de données multidimensionnelles.

1. Ceci est un exemple de graphique construit avec matplotlib.pyplot:

```
abscisses = [1, 2, 3, 4, 5]
ordonnées = [0, 2, -1, 3, 4]
plt.plot(abscisses, ordonnées)
plt.show()
```



2. Réaliser les figures suivantes:



3. Tracer sur un même graphique les courbes des fonctions suivantes :

- $f_1 : x \mapsto 3\sqrt{1-\frac{x^2}{49}}$ sur $[3\,;7]$ et sur $[-7;-3]$
- $f_2 : x \mapsto -3\sqrt{1-\frac{x^2}{49}}$ sur $[4\,;7]$ et sur $[-7;-4]$
- $f_3 : x \mapsto 9-8|x|$ sur $[0.75;1]$ et sur $[-1;-0.75]$
- $f_4 : x \mapsto 0.75+3|x|$ sur $[0.5;0.75]$ et sur $[-0.75;-0,5]$
- $f_5 : x \mapsto 2.25$ sur $[-0.5;0,5]$
- $f_6 : x \mapsto \left|\frac{x}{2}\right| - \frac{3\sqrt{33}-7}{112}x^2 + \sqrt{1-\left(||x|-2|-1\right)^2} - 3$ sur $[-4;4]$
- $f_7 : x \mapsto \frac{6\sqrt{10}}{7} + (1.5-0.5|x|) - \frac{6\sqrt{10}}{14}\sqrt{3+2|x|-x^2}$ sur $[-3;-1]$ et sur $[1;3]$.

4. Cartographie de Données (Heatmap / Corrélation) :
   Générez un jeu de données fictif de 5 variables (A, B, C, D, E). Calculez la matrice de corrélation entre toutes les paires de variables. Visualisez cette matrice de corrélation sous forme d'une carte de chaleur (Heatmap) avec des annotations des coefficients de corrélation.