TD modélisation

Syntaxe:

`np.zeros((n,m))`

Description : fonction créant une matrice (tableau) de taille n×m dont tous les éléments sont nuls.

Argument d'entrée : un tuple de deux entiers correspondant aux dimensions de la matrice à créer.

Argument de sortie : un tableau (matrice) d'éléments de type flottant et égaux à `0`.

Exemple :

Commande	Résultat

`np.zeros((3,4))`	`[[0. 0. 0. 0.]`
	`[0. 0. 0. 0.]`
	`[0. 0. 0. 0.]]`

`A[i,j]`

Description : fonction qui retourne l'élément numéroté `(i,j)` de la matrice A. Pour accéder à l'intégralité de la ligne numérotée i de la matrice A, on écrit `A[i,:]`. De même, pour obtenir toute la colonne numérotée `j` de la matrice A, on utilise la syntaxe `A[:,j]`.

Argument d'entrée : une liste contenant les coordonnées de l'élément dans le tableau `A`.

Arguments de sortie : l'élément appartenant à la ligne numérotée `i` et à la colonne numérotée `j` de la matrice `A`.

RAPPEL : en langage Python, les lignes d'un tableau `A` de taille `n×m` sont numérotées de `0` à `n-1` et les colonnes sont numérotées de `0` à `m-1`.

Exemple :

Commande	Résultat
`A=np.array([[3,4,10],[1,8,7]])`	
`A[0,2]`	`10`
`A[1,:]`	`[1 8 7]`
`A[:,2]`	`[10 7]`

len(l)

argument d'entrée: une liste ou un vecteur l

argument de sortie: le nombre de termes dans la liste ou le vecteur l

Exemple: Commande: l=[0,1,3,7,9,12] Résultat : len(l)=6

`np.sum(a)`

Description : somme tous les éléments de a.

Argument d'entrée : a, une liste de valeurs numériques.

Argument de sortie : somme des valeurs numériques de la liste a.

Exemple `a=[1,2,3]`

Commande	Résultat
`np.sum(a)`	`6`

I. **Moyennes de classe**

Les notes des élèves sont enregistrées dans un tableau de type array nommé notes. Une ligne correspond aux notes obtenues par un élève aux différents devoirs.

Notes=[[12,14,16,11,15],[10,9,6,7,12],[13,12,13,14,15],...]

La note qui est soulignée correspond à Notes[2,1].

1. Ecrire l'instruction 1 qui donne le nombre N d'élèves dans la classe.

2. Ecrire l'instruction 2 qui donne le nombre p de devoirs réalisés.

3. Ecrire l'instruction 3 qui donne un vecteur type array ou un tableau à une ligne nommé MoyDevoirs comprenant les moyennes de la classe aux différents devoirs.

4. Ecrire l'instruction 4 qui donne un vecteur type array ou un tableau à une ligne nommé MoyEleve comprenant les moyennes de tous les élèves lorsque les devoirs ont tous un coefficient 1.

5. Compléter l'instruction 5 qui permet de calculer la meilleure moyenne et le numéro de l'élève qui a obtenu cette meilleure moyenne.

Instruction 5:

max=0

for i in [instruction 5.1]:

——if [instruction 5.2]:

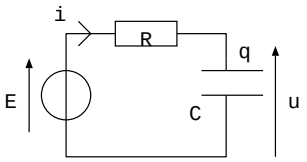
————-max=[instruction 5.3]

————-eleve=[instruction 5.4]

print('la meilleure moyenne est',max)

print('c'est l eleve numero',[instruction 5.5],'qui a obtenu la meilleure moyenne')

II. **Charge d'un condensateur**



On réalise la charge d'un condensateur dans un circuit RC série. On note $i(t)$ l'intensité au cours du temps et $q(t)$ la charge du condensateur.

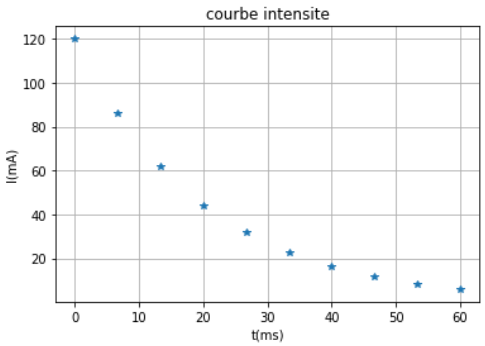
On donne les résultats expérimentaux sous la forme d'un tableau comprenant les couples $(t(ms), i(mA))$.

Tab=[[0,120],[6.6,85.9],[13.3, 61.61],[20,44.1],[26.6,31.63],[33.3,22.66],[40,16.24],[46.6,11.6],[53.3,8.33],[60,5.97]]

1. Donner les valeurs numériques et l'unité de Tab[3,1], Tab[6,0], Tab[9,1]. Compléter le code instruction 1 pour calculer le nombre de couples de mesures.

2. Etablir de façon théorique l'équation différentielle vérifiée par $i(t)$ et en déduire $i(t)$ en fonction de $i(t=0) = I_0$ et d'un temps caractéristique τ.

3. Compléter le code instructions 2 pour tracer la courbe obtenue à partir des mesures. Déduire de cette courbe les valeurs numériques de τ et de I_0.



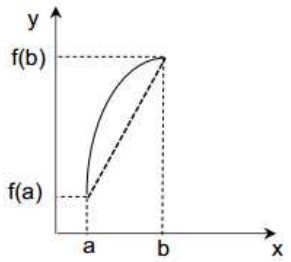
4. Quelle relation existe-t-il entre l'intensité $i(t)$ et $q(t)$? On prend $q(t=0) = 0$, exprimer $q(t)$ en fonction de $i(t)$.

On note I_k la valeur de l'intensité à l'instant t_k. De même, on note $q_k = q(t_k)$.

5. Donner la relation entre q_{k+1}, q_k et $\int_{t_k}^{t_{k+1}} i(t)dt$. Donner l'expression approchée de $\int_{t_k}^{t_{k+1}} i(t)dt$ en fonction de I_k, I_{k+1}, t_k et t_{k+1} à l'aide de la méthode des trapèzes.

Rappel sur la méthode des trapèzes : on cherche à calculer de manière approchée l'intégrale $F = \int_a^b f(x)dx$ *d'une fonction* $f : [a ; b] \to \mathbb{R}$ *continue. La méthode des trapèzes consiste à remplacer* $f(x)$ *sur le segment* $[a ; b]$ *par la fonction affine qui coïncide avec* f *en* a *et* b. *L'intégrale* F *est alors approchée par la formule :*

$$F \approx (b-a)\frac{f(b)+f(a)}{2}.$$



6. Compléter le code, instructions 3, qui permet de calculer une liste nommée listeq avec les valeurs de q_k.

7. On convertit la listeq en vecteur nommé vecq pour pouvoir réaliser des opérations facilement avec les valeurs q_k. Exprimer à tout instant la puissance reçue par le condensateur en fonction de $i(t)$, C et $q(t)$. Compléter le code, instructions 4, pour calculer le vecteur vecP contenant les puissances $P_k = P(t_k)$ et pour calculer la valeur de la puissance maximale $Pmax$ et l'instant $tmax$ pour lequel la puissance est maximale.

```
import numpy as np
import matplotlib.pyplot as plt
# Tableau de mesures
Tab=np.array([[0,120],[6.6,85.9],[13.3, 61.61],[20,44.1],[26.6,31.63],[33.3,22.66],
[40,16.24],[46.6,11.6],[53.3,8.33],[60,5.97]])
N=[instructions 1]
# Tracé de la courbe i(t):
plt.plot([instruction 2.1],'*')
plt.xlabel([instruction 2.2])
plt.ylabel([instruction 2.3])
plt.grid()
plt.title([instruction 2.4])
plt.show()
# Calcul de la charge q(t).
listeq=[instruction 3.1]
for i in range [instruction 3.2]:
——listeq.append([instruction 3.3])
vecq=np.array(listeq)
# Calcul de la puissance reçue par le condensateur
C=10E-5
vecP=[instruction 4.1]
max,indice=0,0
for i in range [instruction 4.2]
—-if [instruction 4.3]:
—— ——max=[instruction 4.4]
—— ——indice=[instruction 4.5]
```