

Etude du temps d'ouverture d'un portail conçu avec un mécanisme 4 barres

Déclaration des variables et appel des bibliothèques :

```
import numpy as np
import matplotlib.pyplot as plt

# variables globales
A=195000
B=70000
C=175000
D=56000
E=140000
F=196000
```

Question 1. Ecrire la fonction f qui a pour arguments les angles θ et β et renvoie la valeur de la fonction f pour un θ et un β donnés.

```
def f(theta,beta):
    """fonction du système 4 barres"""
```

pour $\theta=0.1$ et $\beta=0.5$, $f(0.1,0.5) = 84626.440770428948$

Question 2. Ecrire la fonction $derf_theta$ qui a pour arguments les angles θ et β et qui renvoie la valeur de la fonction dérivée par rapport à θ (β fixé) de $f(\theta, \beta)$ pour un θ et un β donnés.

```
def der_f(theta,beta):
    """dérivée de la fonction par rapport à theta"""
```

Faire un test pour $\theta=0.1$ et $\beta=0.5$, $f'(0.1,0.5) = -257440.06318142801$

Question 3. Ecrire la fonction $newton_theta(f, derive_f, theta0, epsilon, beta)$ qui a pour arguments une fonction f , sa dérivée $derive_f$, la valeur de départ $theta0$, la précision $epsilon$ et la valeur de $beta$ pour le θ recherché. Cette fonction renvoie la solution θ quand la fonction (*) s'annule par la méthode de Newton.

```
def newtonTheta(f, derive_f, theta0, epsilon,beta):
    """methode de Newton avec theta0 la première valeur de résolution avec interruption après 20 itérations"""
    #Initialisation à la première estimation fournie
    theta=theta0
    theta1 =
    while
        theta=theta1
        theta1 =
    return theta1
```

Par la suite, $epsilon=10^{-9}$.

pour $theta0=0$ et $beta=0$ ($\theta = 0.076464006003348267$)

PT - INFO

L'angle β varie dans l'intervalle $[-0.2, 2.4]$ et $\theta_0=0$.

Question 4. Créer la liste `les_beta` de 100 éléments incluant le premier et le dernier élément à pas constant.

```
les_beta=
```

Question 5. Créer la liste `les_theta` correspondant aux angles d'ouverture du portail à partir de la liste `les_beta` et de la fonction `newton_theta`.

```
les_theta=[]
```

Question 6. Tracer la courbe de θ en fonction de β , courbe représentative de la loi entrée/sortie.

```
plt.figure(1)
plt.plot(
plt.xlabel('beta en radian')
plt.ylabel('theta en radian')
plt.title('loi entrée sortie du système 4 barres')
plt.show()
```

On souhaite rechercher dans la liste `les_theta` la position de $\theta = 0$ et $\theta = \frac{\pi}{2}$. La liste `les_theta` ne comporte pas les valeurs exactes 0 et $\frac{\pi}{2}$, nous allons rechercher l'indice de la valeur immédiatement inférieure au θ désiré.

Question 7. Ecrire la fonction `dichotomique(liste, angle)` qui a pour arguments une liste triée et une valeur `angle`. Elle renvoie l'indice de la valeur immédiatement inférieure à `angle` par la méthode de recherche dichotomique.

```
def dichotomique(liste,angle):
    """retourne l'indice de la valeur immédiatement inférieure à angle"""
    g,d=0,len(liste)
    while (d-g)>1:
        m=(d+g)//2
        if liste[m]>:
            else:
    return g
```

`les_theta` est une liste triée de façon croissante.

Question 8. Tester `dichotomique(les_theta,0)` et `dichotomique(les_theta, $\frac{\pi}{2}$)`. On appellera les indices renvoyés i_0 et i_{90} . Vérifier que les valeurs de θ correspondant aux indices i_0 et i_{90} sont proches de 0 et $\frac{\pi}{2}$.

Question 9. Donner les deux valeurs de β correspondant aux indices i_0 et i_{90} .

PT - INFO

Question 10. Connaissant la vitesse de rotation du bras moteur $\omega_{bras} = \dot{\beta} = 0,1 \text{ rad/s}$, calculer le temps d'ouverture du vantail.

```
def temps(beta0,beta1):  
    return
```

On donne $\Delta t = \frac{les_beta[i_0]-les_beta[i_0+1]}{\omega_{moteur}}$ avec i_0 l'indice obtenu pour $\theta = 0 \text{ rad}$.

Question 11. Définir la liste *les_theta_tronquee* à partir de *les_theta[i_0]* jusqu'au *les_theta[i_90+1]*

```
i0=dichotomique(les_theta,0)  
i90=dichotomique(les_theta,np.pi/2)  
les_theta_tronquee=les_theta[
```

Question 12. A partir de la liste *les_theta_tronquee* et *delta_t*, créer une liste *les_omega* des vitesses en rad/s

pour $i = 0, \omega(0) = 0$ et $\theta(0) = 0$

$$\text{pour } i \geq 1, \omega_i = \frac{\theta_i - \theta_{i-1}}{\Delta t}$$

```
delta_t=(les_beta[1]-les_beta[0])/0.1  
les_omega=[0]  
for i in range(len(les_theta_tronquee)-1):  
    les_omega.append(  
plt.figure(2)  
plt.plot(les_theta_tronquee,les_omega)  
plt.show()
```

Question 13. Définir la fonction *rechercheMax(liste)* qui renvoie la valeur maximale d'une liste.

```
def rechercheMax(liste):  
    M=liste[0]  
    for i in range(len(liste)):  
  
    return (M)
```

Question 14. Déterminer la vitesse de rotation maximale du vantail

```
vitesse=
```

Question 15. Ouvrir et lire avec python le fichier *modeles_portail.csv* donnant les dimensions de 4 modèles de portail. Afficher la ligne du modèle 1.

Si vous ouvrez par ailleurs le fichier dans un tableur ou un éditeur de texte, n'oubliez pas de le fermer avant de compiler votre programme python.

```
fichier=open('modeles_portail.csv','r')  
  
# print (lignes[1])
```

PT - INFO

Question 16. Créer 6 listes notées *liste_A*, *liste_B*... d'entiers contenant respectivement les valeurs des constantes A, B ... des quatre modèles.

```
les_A=[]
les_B=[]
les_C=[]
les_D=[]
les_E=[]
les_F=[]
for i in range(1,len(lignes)):
    l=lignes[i].strip()
    l=l.split(';')
    les_A.append(
    les_B.append(
    les_C.append(
    les_D.append(
    les_E.append(
    les_F.append(
```

Question 17. Analyser et comprendre le code permettant de déterminer le temps t_{90} d'ouverture du portail et la vitesse maximale de rotation du vantail.

```
duree=[]
vitesse=[]
delta_t=(les_beta[1]-les_beta[0])/0.1
for i in range(4):
    A,B,C,D,E,F=les_A[i],les_B[i],les_C[i],les_D[i],les_E[i],les_F[i]
    les_theta=[]
    for b in les_beta:
        les_theta.append(newtonTheta(f,der_f,0,10**-9,b))
    i0=dichotomique(les_theta,0)
    i90=dichotomique(les_theta,np.pi/2)
    t=temps(les_beta[i0],les_beta[i90])
    duree.append(t)
    les_theta_tronquee=les_theta[i0:i90+1]
    les_omega=[0]
    for i in range(len(les_theta_tronquee)-1):
        les_omega.append((les_theta_tronquee[i+1]-les_theta_tronquee[i])/delta_t)
    vitesse.append(rechercheMax(les_omega))
```

Question 18. Ecrire dans un fichier *exo2_nom_prenom.csv* les valeurs obtenues sous la forme :

Type de modèle	Temps ouverture	Vitesse maxi
Modèle 1		
Modèle 2		

```
fichier2=open('synthese_prof_vitesse.csv','w')
fichier2.write
modeles=['modele 1','modele 2','modele 3','modele 4']
for i in range(4):
    fichier2.write(
fichier2.close()
```