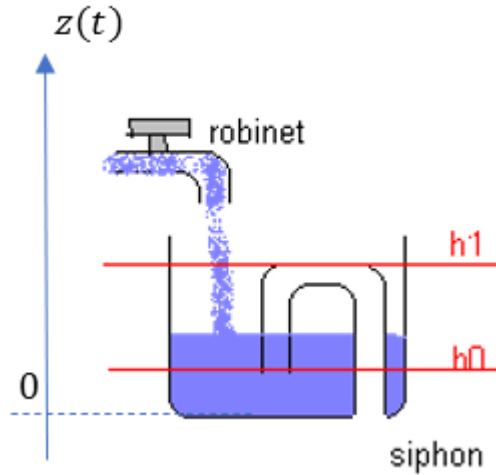


Le vase de Tantale : d86f-6558269

Le vase de tantale est le dispositif ci-dessous :



I- Le 1e remplissage

Le récipient est initialement vide et le robinet est ouvert assurant ainsi un débit massique D constant. Le siphon n'est pas encore amorcé et aucune vidange n'est alors possible.

1) Mener un bilan de la la masse $m(t)$ contenue dans le récipient et donner l'expression de $\frac{dm(t)}{dt}$.

1. Tracer l'allure de $z(t)$ où z est la hauteur de fluide dans le récipient. On note S la section du récipient et ρ la masse volumique du fluide.

II- vidange

Lorsque la hauteur de fluide atteinte la valeur h_1 le siphon est amorcée et une vidange commence. On ne s la section du siphon. Dans la suite, on suppose le fluide parfait, incompressible en écoulement quasisationnaire car $S \gg s$.

1. En appliquant Bernoulli sur une ligne de courant adaptée, en déduire une expression du débit D' associée à la vidange par le siphon en fonction de s , g et $z(t)$ où g est l'intensité du champ de pesanteur terrestre.

1. Montrez alors que :

$$\frac{dz}{dt} = \frac{D - \rho s \sqrt{2gz(t)}}{\rho S}$$

1. Quelle condition sur D assure une vidange ?

1. On va supposer que $D \ll D'$. Déterminer alors le temps de vidange.

III- Reprise du remplissage

La vidange précédente conduit au désamorçage du siphon lorsque $z = h_0$.

1. Déterminer le temps pour remplir le récipient de h_0 à h_1 .

1. En déduire l'expression de la période des oscillations

IV- Simulation

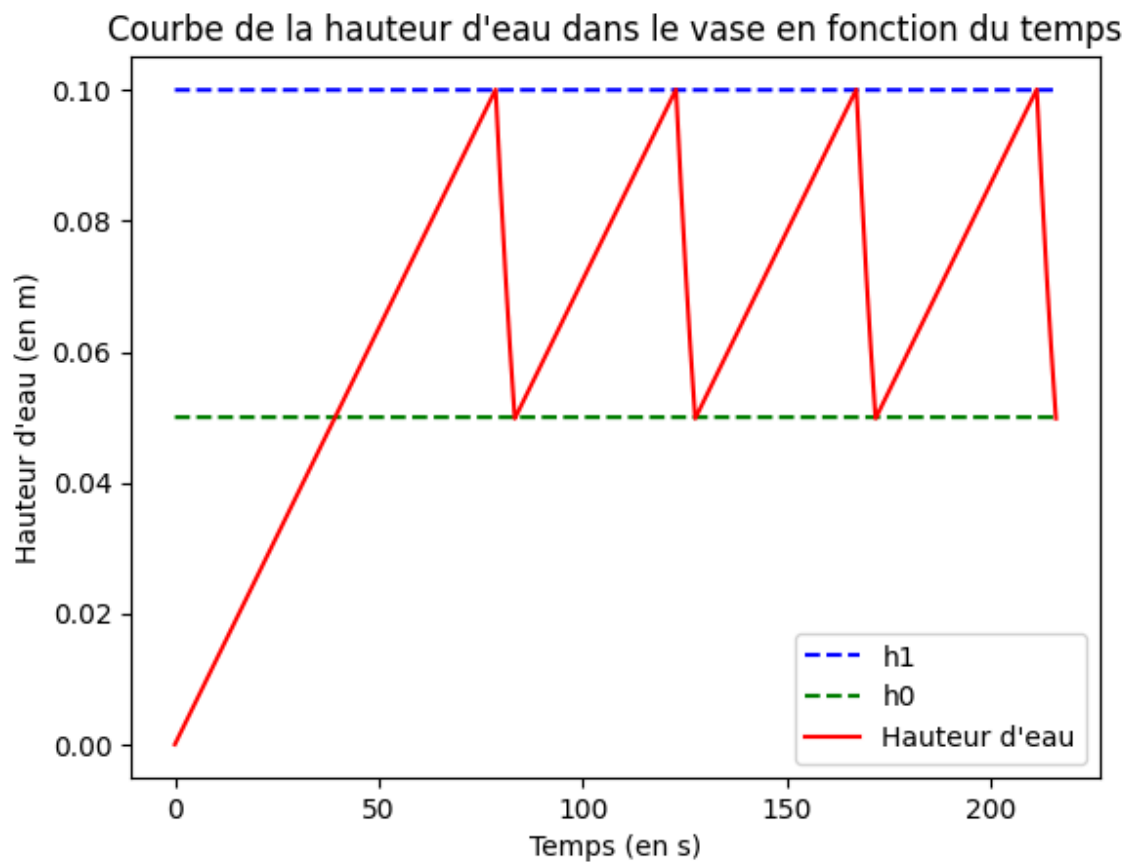
Compléter les 2 instructions incomplètes :


```

In [16]: 1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 D = 0.1
5 d = 0.01
6 S=np.pi*(D/2)**2
7 s=np.pi*(d/2)**2
8 h0 = 0.05
9 h1 = 0.1
10 Dm = 10**-2
11 g = 9.8
12 rho=10**3
13
14 "Paramètres informatiques"
15 Te = 0.1 #Pas de discrétisation (en seconde)
16 temps = [0] #Liste des temps
17 z = [0] #Liste des hauteurs d'eau
18 Ncycle = 3 #Nombre de période de l'oscillateur à afficher
19
20 "Valeurs courantes des paramètres à regarder"
21 zc = 0
22 tc = 0
23 Ncyclec = 0
24
25 def f1(z):
26     return Dm/(S*rho)
27
28 def f2(z):
29     return Dm/(S*rho) - s*np.sqrt(2*g*z)/(S)
30
31 "Résolution des équations différentielles par la méthode d'Euler ex
32 while Ncyclec <= Ncycle :
33     "Phase 1 : Remplissage"
34     while zc < :#à compléter!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
35         tc = tc + Te
36         zc = zc + f1(zc)*Te
37         temps.append(tc)
38         z.append(zc)
39     "Phase 2 : Vidange"
40     while zc > :#à compléter!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
41         tc = tc + Te
42         zc = zc + f2(zc)*Te
43         temps.append(tc)
44         z.append(zc)
45     Ncyclec += 1
46
47 "Affichage des courbes"
48 N = len(z)
49 h1liste = [h1]*N
50 h2liste = [h0]*N
51 plt.plot(temps,h1liste,'--b',label="h1")
52 plt.plot(temps,h2liste,'--g',label="h0")
53 plt.plot(temps,z,'r',label="Hauteur d'eau")
54 plt.xlabel("Temps (en s)")
55 plt.ylabel("Hauteur d'eau (en m)")
56 plt.title("Courbe de la hauteur d'eau dans le vase en fonction du t
57 plt.legend()
58 plt.show()
59
60

```

61



In []:

1