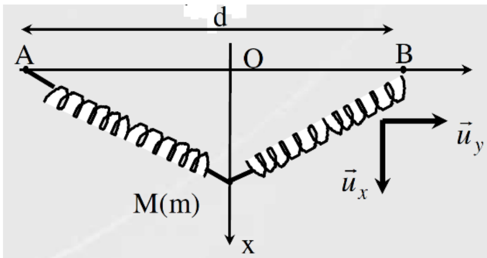
Trampoline

Question simple



On considère la modélisation d'un trampoline à l'aide de deux ressorts de longueur à vide ℓ_0 et de raideur k. Un trampoliniste, assimilé à un point matériel M de masse m monte sur le trampoline qui s'enfonce ; son mouvement est vertical le long de l'axe (Ox).

Données : $k = 10000\ N \cdot m^{-1}$; $\ell_0 = 1{,}0\ m$; $m = 80\ kg$; $d = 4{,}5\ m$

1) **Etablir l'expression de l'énergie potentielle du point matériel M. Etablir alors une équation dont la résolution permet de déterminer la valeur de la distance d'enfoncement x_{eq} à l'équilibre lorsque l'homme monte sur le trampoline.**

2) L'équation précédente est résolue avec la fonction *bisect* avec python dont on donne ci-dessous la spécificité.

La commande `bisect` opère par dichotomie. Sa syntaxe s'écrit `bisect(fonction,borne_inf,borne_sup)`.

Pour que celle-ci fonctionne, il est nécessaire de :

- définir préalablement une fonction $f(x)$,
- préciser un intervalle $[a, b]$ à l'intérieur duquel rechercher la racine de f,
- s'assurer que $f(a)$ et$f(b)$ sont de signes opposés.

Le script est le suivant :

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import scipy.optimize as op
4  
5  #Paramètres
6  g=9.81 #m/s^2
7  m=80 #kg
8  k=10000 #N/m
9  l0=1 #m
10 d=4.5 #m
11 
12 def equilibre(x):
13     return
14 
15 xeq=op.bisect(equilibre,0,1)
16 
17 print(xeq)
18 
```

Compléter la ligne 13 du script

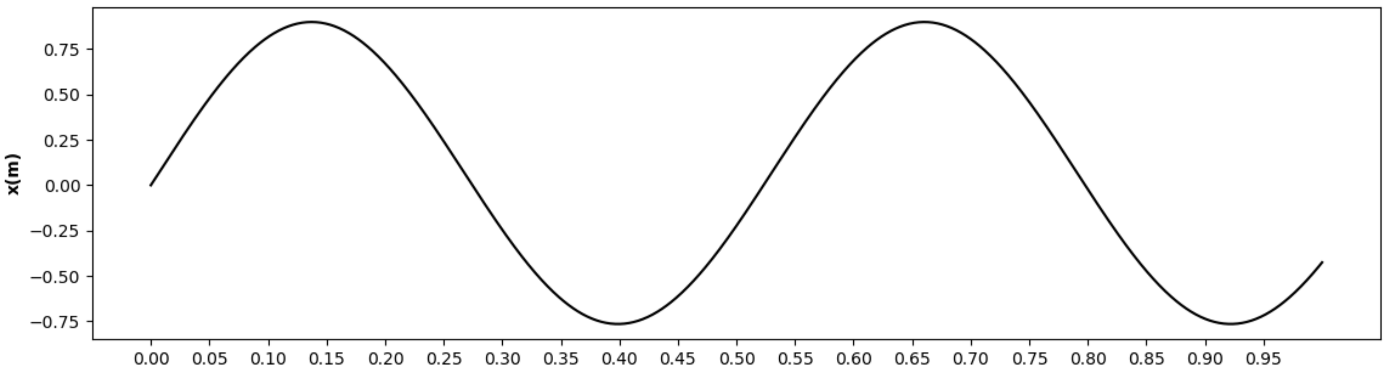
Question ouverte

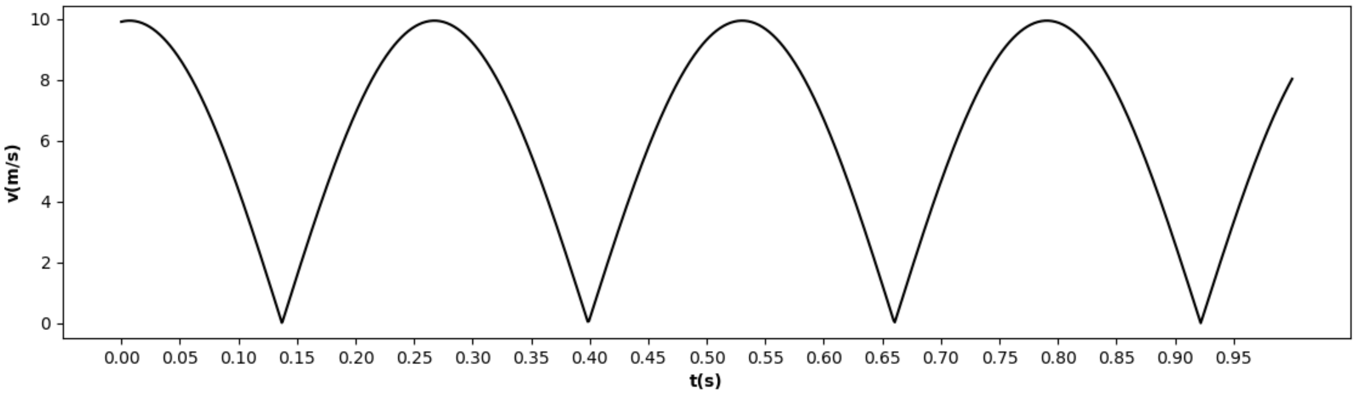
Le trampoliniste est à présent en action. Après plusieurs sauts, il atteint une hauteur constante et égale à 5,0 m.

Estimer la période d'un saut de l'athlète

On s'aidera du script suivant, après notamment avoir complété les lignes 14, 26 et 28, qui permettent de résoudre l'équation du mouvement et de tracer les courbes d'évolution dans le temps de la position et de la vitesse du sportif lorsqu'il est en contact avec le trampoline.

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy.integrate import odeint
4  
5  #Paramètres
6  g=9.81 #m/s^2
7  m=80 #kg
8  k=10000 #N/m
9  l0=1 #m
10 d=4.5 #m
11 hmax=5 #m
12 
13 #Résolution de l'équation différentielle du mouvement
14 v0=                 #vitesse initiale lorsque l'athlète entre en contact avec le
15                     #trampoline
16 x0=0
17 F0=x0,v0
18 
19 t0,tmax=0,1#s
20 N=1000
21 t=np.linspace(t0,tmax,N)
22 
23 def derive(F,t):
24     x=F[0]
25     xprim=F[1]
26     return np.array([                                              ])
27 
28 solution=odeint(                          )
29 x=solution[:,0]
30 v=abs(solution[:,1])
31 
32 #Tracés de x et v en fonction du temps
33 plt.subplot(211)
34 plt.plot(t,x,'k')
35 plt.axhline(y=0,linewidth=1,linestyle='--',color='k')
36 plt.ylabel("x(m)",fontweight='bold')
37 
38 plt.subplot(212)
39 plt.plot(t,v,'k')
40 plt.xlabel("t(s)",fontweight='bold')
41 plt.ylabel("v(m/s)",fontweight='bold')
42 plt.show()
```





Dans le script la fonction ***odeint*** est utilisée dont la spécificité est donnée ci-dessous :

L'intégration de l'équation différentielle est ici réalisée grâce à la fonction `odeint` qui se trouve dans le module `integrate` de la bibliothèque `scipy`.
Lorsque l'équation est du second ordre en une variable **u**, la résolution nécessite de créer un vecteur contenant la grandeur **u** et sa dérivée première $\frac{du}{dt}$:

$$X = \begin{pmatrix} u \\ \frac{du}{dt} \end{pmatrix}$$

Il faut ensuite définir une fonction **f** qui dérive le vecteur le vecteur X :

$$f : \begin{pmatrix} u \\ \frac{du}{dt} \end{pmatrix} \mapsto \begin{pmatrix} \frac{du}{dt} \\ \frac{d^2u}{dt^2} \end{pmatrix}$$

Lors de la définition de la fonction **f**, le terme de dérivée seconde est remplacé par l'expression issue de l'équation différentielle.

Exemple avec l'équation différentielle précédemment décrite :

$$\frac{d^2u}{dt^2} = g\left(\frac{du}{dt}\right) + h(u)$$

$$f : \begin{pmatrix} u \\ \frac{du}{dt} \end{pmatrix} \mapsto \begin{pmatrix} \frac{du}{dt} \\ \left(\frac{du}{dt}\right) + h(u) \end{pmatrix}$$

Trois arguments doivent être saisis dans la commande `odeint` pour que l'intégration soit réalisée :

```
odeint(f, X0 , t)
```

où :

- `X0` désigne la liste des conditions initiales u_0 et $\left(\frac{du}{dt}\right)_0$
- `t` désigne la liste des instants t. Celle-ci peut créée grâce à la fonction `linspace` de la bibliothèque `numpy` sous la forme `np.linspace(borne_inf,borne_sup,nombre_valeurs)`.

`odeint` renvoie un tableau dont la première colonne est la liste des valeurs successives de **u** et la seconde, celles des valeurs succesives de la dérivée première $\frac{du}{dt}$.

Pour extraire les listes utiles de X, :

- liste des valeurs de **u** : `X[:,0]` (1ère colonne)
- liste des valeurs de $\frac{du}{dt}$: `X[:,1]` (2ème colonne)

Correction

Question simple :

Référentiel : terrestre supposé galiléen

Système : trampoliniste assimilé à un point matériel M de masse m

Forces extérieures :

- Poids : $\vec{P}$
- Force de rappel exercée par le ressort de gauche : $\overrightarrow{F_g}$
- Force de rappel exercée par le ressort de droite : $\overrightarrow{F_g}$

L'énergie potentielle de ce système s'écrit : $E_p = E_{pp} + E_{peg} + E_{ped} = -mgx + \frac{1}{2}k\left(l_g - l_0\right)^2 + \frac{1}{2}k(l_d - l_0)^2$

$$\text{avec } l_g = l_d = \sqrt{x^2 + \left(\frac{d}{2}\right)^2}$$

On en déduit : $E_p = E_{pp} + E_{pe} = -mgx + k\left(\sqrt{x^2 + \left(\frac{d}{2}\right)^2} - l_0\right)^2$ (à une constante près)

A l'équilibre : $\vec{P} + \overrightarrow{F_g} + \overrightarrow{F_g} = \vec{0} \Leftrightarrow \left(\frac{dE_p}{dx}\right)_{x=x_{eq}} = 0 \Rightarrow -mg + 2k \times \left(\frac{2x_{eq}}{2\sqrt{x_{eq}{}^2+\left(\frac{d}{2}\right)^2}}\right) \times \left(\sqrt{x_{eq}{}^2 + \left(\frac{d}{2}\right)^2} - l_0\right) = 0$

$$\Rightarrow -mg + 2kx_{eq} \times \left(1 - \frac{l_0}{\sqrt{x_{eq}{}^2 + \left(\frac{d}{2}\right)^2}}\right) = 0$$

Dans la **ligne 13** du script python, il faut donc écrire en langage python : $-mg + 2kx \times \left(1 - \frac{l_0}{\sqrt{x^2+\left(\frac{d}{2}\right)^2}}\right)$

Question ouverte :

Période d'un saut = temps passé en l'air + temps passé sur le trampoline avec un retour au point de départ :

$$T = t_{air} + t_{trampo}$$

Calcul de t_{air} : $t_{air} = t_{air,d} + t_{air,a}$ (phase descendante puis ascendante)

Hypothèse : système soumis uniquement au poids dans l'air

En phase descendante (depuis les $h_0 = 5m$ où l'athlète est sans vitesse, pris comme origine des temps)

PFD : $m\vec{a} = m\vec{g} \Rightarrow \vec{a} = \vec{g}$ en projetant suivant l'axe (Ox) : $\ddot{x} = g \Rightarrow \dot{x} = gt + 0 \Rightarrow x = \frac{1}{2}gt^2 - h_0$

$x\left(t_{air,d}\right) = 0 \Rightarrow \frac{1}{2}gt_{air,d}{}^2 - h_0 = 0 \Rightarrow t_{air,d} = \sqrt{\frac{2h_0}{g}}$ AN : $t_{air,d} = 1{,}0\ s$

La vitesse d'arrivée sur le trampoline (qui peut aussi être obtenue par le TEM) est :

Ligne 14 $v_0 = g \times t_{air,d} = \sqrt{2gh_0} = 9{,}9\ m \cdot s^{-1}$ confirmé par le graphe (vitesse initiale)

En phase ascendante et en absence de forces dissipatives, la vitesse à la sortie du trampoline sera celle à l'entrée (ce que confirme le graphe donnant la vitesse en fonction du temps).

PFD (projeté suivant l'axe Ox et en prenant comme origine des temps l'instant du décollage : $\dot{x}(t=0) = -v_0$ et $x(t=0) = 0$: $\ddot{x} = g \Rightarrow \dot{x} = gt - v_0 \Longrightarrow x = \frac{1}{2}gt^2 - v_0 t$

$$x(t_{air,a}) = -h_0 \Longrightarrow \frac{1}{2}g{t_{air,a}}^2 - v_0 t_{air,a} + h_0 = 0$$

La résolution de l'équation du second degré donné $t_{air,a} = t_{air,d} = 1{,}0\ s$ ce qu'on peut aussi donner directement

Dans la phase où l'athlète est en contact avec le trampoline, et toujours en absence de forces dissipatives, les forces présentes sont le poids et les deux forces de rappel, le système est conservatif, l'énergie mécanique se conserve au cours du temps :

$E_m = E_c + E_p = E_c + E_{pp} + E_{pe} = \frac{1}{2}mv^2 - mgx + k\left(\sqrt{x^2 + \left(\frac{d}{2}\right)^2} - l_0\right)^2 = \frac{1}{2}m\dot{x}^2 - mgx + k\left(\sqrt{x^2 + \left(\frac{d}{2}\right)^2} - l_0\right)^2$ (à une constante près) (voir question simple)

$$\frac{dE_m}{dt} = 0 \Longrightarrow m\dot{x}\ddot{x} - mg\dot{x} + 2k \times \left(\frac{2\dot{x}x}{2\sqrt{x^2 + \left(\frac{d}{2}\right)^2}}\right) \times \left(\sqrt{x^2 + \left(\frac{d}{2}\right)^2} - l_0\right) = 0$$

$$\Longrightarrow\ m\ddot{x} - mg + 2kx \times \left(1 - \frac{l_0}{\sqrt{x^2 + \left(\frac{d}{2}\right)^2}}\right) = 0 \Longrightarrow \ddot{x} = -g + \frac{2k}{m}x \times \left(1 - \frac{l_0}{\sqrt{x^2 + \left(\frac{d}{2}\right)^2}}\right)$$

Ligne 26 $x_{prim}\ , -g + \frac{2k}{m}x \times \left(1 - \frac{l_0}{\sqrt{x^2 + \left(\frac{d}{2}\right)^2}}\right)$

Ligne 28 *derive, F0, t*

Le temps passé sur le trampoline se trouve sur les graphes : ce temps correspond au premier temps pour lequel $x = 0$ à nouveau (point de décollage), on lit : $t_{trampo} \approx 0{,}27\ s$.

La période vaut donc : $\boldsymbol{T} \approx 2{,}27\ \boldsymbol{s}$ valeur trop élevée par rapport à la réalité

Critiques possibles : forces de frottements solide et dans l'air, impulsion de l'athlète