

Devoir en temps libre n° 22

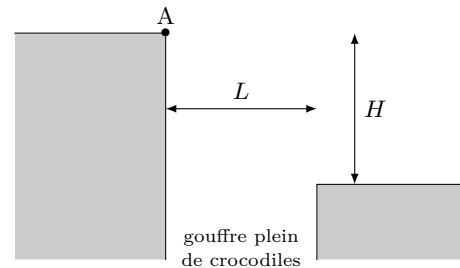
Le saut de la mort

Superlapin, poursuivi par Evilfox, doit franchir un précipice d'une largeur de $L = 10\text{ m}$ dont l'autre bord se trouve $H = 2\text{ m}$ plus bas que celui où il se trouve. On admet que Superlapin court à $v_0 = 40\text{ km} \cdot \text{h}^{-1}$ et que sa vitesse est conservée lorsqu'il entame son saut au point A ; seule la direction de sa vitesse change.

Déterminer l'intervalle dans lequel doit se trouver l'angle avec lequel il saute (compté par rapport à l'horizontale) pour échapper à la mort.

On commencera par mettre le problème en équation, ou on utilisera une ou plusieurs équations établies en cours, à condition de préciser clairement en quoi elle permet de résoudre le problème proposé.

On fera une résolution graphique ou numérique à l'aide d'un programme Python.



Devoir en temps libre n° 22

éléments de correction

Il s'agit d'un problème analogue au tir d'un projectile, traité en cours. En prenant comme système Superlapin, et une base $(A, \vec{u}_x, \vec{u}_z)$ centrée au point origine du saut, sous l'effet du seul poids et avec une vitesse initiale $\vec{v}_0$ inclinée d'un angle α par rapport à l'horizontale, on obtient facilement :

$$\vec{a} = \begin{cases} 0 \\ -g \end{cases} \Rightarrow \vec{v} = \begin{cases} v_0 \cos \alpha \\ -gt + v_0 \sin \alpha \end{cases} \Rightarrow \overrightarrow{AM} = \begin{cases} x = v_0 \cos \alpha t \\ z = -gt^2/2 + v_0 \sin \alpha t \end{cases}$$

L'équation cartésienne de la trajectoire s'obtient en éliminant le temps dans ce système d'équations :

$$t = \frac{x}{v_0 \cos \alpha} \quad \text{et} \quad z = -\frac{g}{2} \left(\frac{1}{v_0 \cos \alpha} \right)^2 \times x^2 + \tan \alpha \times x$$

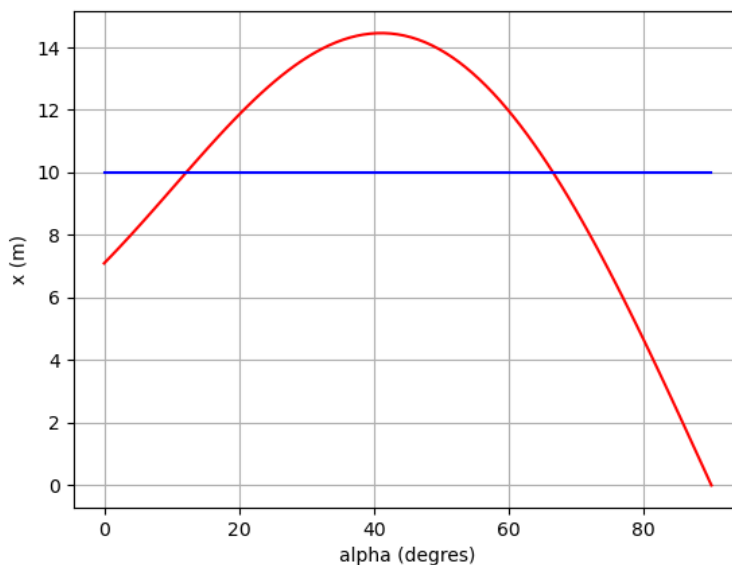
Superlapin s'en sort s'il s'est déplacé d'au moins L selon la direction horizontale (selon $\vec{u}_x$) lorsqu'il est descendu de H suivant la verticale (selon $-\vec{u}_z$). Autrement dit, il faut que $x \geq L$ lorsque $z = -H$. Or, les valeurs de x atteintes pour $z = -H$ sont solution du polynôme du second degré :

$$-H = -\frac{g}{2} \left(\frac{1}{v_0 \cos \alpha} \right)^2 \times x^2 + \tan \alpha \times x \Rightarrow \underbrace{\frac{g}{2} \left(\frac{1}{v_0 \cos \alpha} \right)^2}_{=a} \times x^2 - \underbrace{\tan \alpha}_{=b} \times x - H = 0$$

La valeur de x atteinte pour $z = -H$ ayant un sens physique (celle qui est positive) est :

$$x = \frac{-b + \sqrt{\Delta}}{2a} \quad \text{avec} \quad \Delta = b^2 - 4 \times a \times (-H)$$

Il suffit donc de déterminer la valeur de x atteinte pour $z = -H$ et pour chaque valeur de α , autrement dit de résoudre le polynôme précédent pour chaque valeur de α , et comparer la valeur de x trouvée avec L . La résolution peut se faire à l'aide d'un programme Python. Les angles qui permettent à Superlapin de triompher sont compris entre $\alpha_{\min} = 12,5^\circ$ et $\alpha_{\max} = 66,5^\circ$.



```
1 # Importation des bibliotheques
2
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from math import pi
6
7
8 # Definition des constantes
9
10 v0 = 11.11
11 g = 9.8
12 H = 2      # hauteur du sol apres le precipice
13 L = 10     # longueur minimale a sauter
14
15
16 # Creation des tableaux numpy necessaires
17 # Tableau contenant tous les angles alpha entre 0 et 90 par pas de 0,5 degre
18 alpha = np.linspace(0,90,181)
19
20 # Tableaux des valeurs des coefficients a et b du polynome et du discriminant
21 a = np.array([])
22 b = np.array([])
23 delta = np.array([])
24
25 # Tableaux des solutions x du polynome et des alpha tels que  $x \geq L$ 
26 x = np.array([])
27 alpha_sauve = []
28
29
30 # Remplissage des tableaux par resolution du polynome pour chaque valeur de alpha
31 # Attention a convertir les valeurs de alpha en radian pour calculer cos et tan
32 a = g/2/v0**2/(np.cos(alpha*pi/180)**2)
33 b = -np.tan(alpha*pi/180)
34 delta = b**2-4*a*(-H)
35 x = (-b+np.sqrt(delta))/2/a
36
37
38 # Trace de x en fonction de alpha (rouge) et trace de la valeur limite L (bleu)
39 plt.figure(1)
40 plt.plot(alpha,x,'r-')
41 plt.plot(alpha, L*np.ones(len(alpha)), 'b-')
42 plt.xlabel('alpha (degres)')
43 plt.ylabel('x (m)')
44 plt.grid()
45 plt.show()
46
47
48 # Remplissage du tableau des valeurs de x qui sauvent Superlapin
49 # On garde les valeurs de alpha telles que  $x \geq L$ 
50 i = 0
51 while i < len(alpha) :
52     if x[i] >= L :
53         alpha_sauve.append(alpha[i])
54     i = i+1
55
56
57 # Affichage des deux valeurs extreme de alpha_sauve
58 print('alpha_min = ', alpha_sauve[0] , ' degres')
59 print('alpha_max = ', alpha_sauve[len(alpha_sauve)-1] , ' degres')
```