

3 Bilan énergétique

Script à compléter

L'objectif de cette activité est d'étudier le mouvement de glissement d'un système sur un plan incliné, avec ou sans frottements.

Fichiers Python

Script à compléter
Fiche d'accompagnement
hatier-clic.fr/pc1310

Prérequis théoriques

- Conservation ou non de l'énergie mécanique
- Travail d'une force de frottements

Apports théoriques supplémentaires de l'activité

- Loi de Coulomb de frottements, simplifiée

La question 1a nécessite les **modifications du programme** suivantes :

- l'utilisation de la conservation de l'énergie mécanique pour calculer l'énergie mécanique en B (ligne 89 du script de la page suivante) ;
- l'écriture de l'expression de l'énergie potentielle de pesanteur en B (ligne 102) ;
- l'écriture de l'expression de l'énergie cinétique en B, calculée à l'aide de l'énergie mécanique et de l'énergie potentielle de pesanteur (ligne 103) ;
- l'écriture de l'expression de la vitesse en B, en fonction de l'énergie cinétique en B et de la masse (ligne 104).

Les questions 1b et 1c consistent en l'exécution du programme et l'étude des valeurs calculées de la vitesse en B, **en faisant varier les valeurs** de la vitesse initiale (ligne 17), de la longueur de la piste (ligne 15), de la masse du système (ligne 14).

La question 2 implique des **modifications du programme** pour prendre en compte les frottements :

- un test pour savoir s'il y a glissement ou non, conformément à l'indication du **doc. 2** relative à l'angle d'inclinaison du plan (lignes 73 et 74) ;
- l'écriture de l'expression de la norme de la force de frottement en utilisant l'expression du **doc. 2** (ligne 93) ;
- l'écriture de l'expression du travail de ces forces, vue la première fois au **chapitre 13** (ligne 96) ;
- l'écriture de l'expression de l'énergie mécanique en B, en utilisant le théorème de l'énergie mécanique (ligne 99).

Le programme ainsi réalisé doit être testé avec diverses valeurs des paramètres initiaux, dont le coefficient de frottement μ (ligne 18).

Script à compléter

```

1 from math import *
2 import sys
3
4 print("")
5 print("*****")
6 print("*          Bilan énergétique          *")
7 print("* système en glissement sur une pente *")
8 print("*****")
9 print("")
10
11 ### Données ###
12 g=9.8 # norme du champ de pesanteur en newtons par kilogramme
13 yB=0 # altitude du point d'arrivée B en mètres
14 m=1 # masse du système en kilogrammes
15 L=20 # longueur de la piste en mètres
16 yA=10 # altitude du point de départ A en mètres
17 vA=0 # vitesse du système au départ en mètres par seconde
18 mu=0.2 # coefficient de frottement
19
20 ### Tests d'interruption du programme ###
21 problemes=[]
22 if L<=0:
23     problemes.append("L doit être strictement positive")
24 if m<=0:
25     problemes.append("m doit être strictement positive")
26 if vA<0:
27     problemes.append("vA ne peut pas être négative")
28 if g!=9.8:
29     problemes.append("g doit rester à 9,8 N/kg")
30 if yB!=0:
31     problemes.append("yB doit rester nulle")
32 if L<yA-yB:
33     problemes.append("L et le dénivelé sont incompatibles")
34 if yA<yB:
35     problemes.append("A est au-dessous de B")
36 if mu<0:
37     problemes.append("mu est négatif")
38 if mu!=0 and vA!=0:
39     problemes.append("vA n'est pas nulle mais il y a des frottements")
40 if len(problemes)!=0:
41     for i in range(len(problemes)):
42         print(problemes[i])
43     sys.exit("Modifiez les paramètres et réessayez")
44
45 ### Fonction d'affichage des énergies en un point ###
46 def affiche_energies(point, Ec, Epp, Em):
47     print("Valeurs des énergies du système en", point, ":")
48     print("- Énergie cinétique :", round(Ec, 1), "J")
49     print("- Énergie potentielle de pesanteur :", round(Epp, 1), "J")
50     print("- Énergie mécanique :", round(Em, 1), "J")
51     print("")
52
53 ### Affichage des données ###
54 print("Données")
55 print("- Masse du système :", m, "kg")
56 print("- Longueur de la piste :", L, "m")
57 print("- Altitude du point de départ :", yA, "m")
58 print("- Altitude du point d'arrivée :", yB, "m")
59 print("- Vitesse initiale du système :", vA, "m/s")
60 print("- Coefficient de frottement :", mu)
61 print("")

```

Modules

Le module `maths` est nécessaire pour les fonctions trigonométriques et la fonction racine carrée qui peut être utilisée pour le calcul de la vitesse.

Le module `sys` fournit `sys.exit()` qui permet l'interruption du programme.

Données à modifier

Question 1b : la valeur de y_A

Question 1c : les valeurs de v_A , L et m

Données à ne pas modifier : g et y_B

Attention Le séparateur décimal est le point, pas la virgule.

Tests (ne pas modifier)

Cette partie du programme vérifie le réalisme et la cohérence des données. Le programme impose notamment que B soit à l'altitude y_B nulle et que A soit en dessous de B. L'activité se restreignant au cas d'un glissement avec frottements mais sans vitesse initiale, cette incompatibilité est aussi testée. On peut s'inspirer de la syntaxe de cette partie pour le test du glissement ou non dans la question 2 : la fonction `sys.exit("Message")` interrompt le programme et affiche le message. Il est nécessaire pour l'utiliser d'importer le module `sys`.

```

62
63 ### Angle alpha de la piste au-dessus de l'horizontale ###
64 alpha=asin(float((yA-yB)/L))
65 alphadegre=alpha*180/pi
66 print("Angle entre la piste et l'horizontale :",round(alphadegre,1),"°")
67 print("")
68
69 #####
70 ##### Question 2 - à compléter #####
71 ### test du glissement et interruption ###
72 #####
73
74
75
76 ### Calcul et affichage des énergies en A ###
77 EcA=0.5*m*vA**2
78 EppA=m*g*yA
79 EmA=EcA+EppA
80 affiche_energies("A",EcA,EppA,EmA)
81
82 #####
83 ##### À compléter : question 1a puis question 2 #####
84 ### Utilisation du théorème de l'énergie mécanique ###
85 #####
86 if mu==0:
87     print("Hypothèse de conservation de l'énergie mécanique")
88     ### Calcul de EmB
89
90 else:
91     print("Hypothèse de la loi de frottement de Coulomb")
92     ### Calcul et affichage de la norme des frottements
93
94     print("- Norme de la force de frottements :",round(f,1),"N")
95     ### Calcul et affichage du travail des frottements
96
97     print("- Travail de la force de frottements :",round(Wf,1),"J")
98     ### Calcul de EmB
99
100
101 ### Calcul des énergies et de la vitesse en B
102
103
104
105
106 ### Affichage des valeurs finales ###
107 print("")
108 affiche_energies("B",EcB,EppB,EmB)
109 print("Valeur de la vitesse en B :",round(vB,1),"m/s")
110 print("")

```

Calcul et affichage de l'angle (ne pas modifier)

Les fonctions trigonométriques utilisent par défaut les radians. Pour l'affichage, l'angle est converti en degrés. Les valeurs calculées sont arrondies à un chiffre après la virgule pour affichage par `round(valeur,1)`.

Question 2 : test de glissement

Le **doc. 2** indique que si $\tan(\alpha)$ est inférieur au coefficient de frottement μ , il n'y a pas de glissement. La fonction `sys.exit("Message")` interrompt le programme et affiche le message. Il est nécessaire pour l'utiliser d'importer le module `sys`.

Théorème de l'énergie mécanique (questions 1a et 2)

Le théorème de l'énergie mécanique permet de calculer l'énergie mécanique en B. Dans le cas avec frottements (question 2), cela nécessite de connaître la norme des frottements, qui est donnée par le **doc. 2** de l'activité, et de calculer son travail.

Point B : calculs de l'énergie potentielle de pesanteur, de l'énergie cinétique, puis de la vitesse (question 1a)

On utilise les expressions du cours pour calculer ces grandeurs. La fonction racine carrée, nécessaire pour calculer la vitesse en B, est soit `sqrt(valeur)` qui nécessite l'import du module `maths`, soit `valeur**0.5`.