

### 3 Peser un astre

#### Programme à compléter

L'objectif de cette activité est d'exploiter des données satellitaires ou astronomiques pour tester la troisième loi de Kepler.

#### Fichiers Python

Programme à compléter  
Fiche d'accompagnement  
[hatier-clic.fr/psi378](http://hatier-clic.fr/psi378)

#### Prérequis théoriques

- Aucun prérequis réel, les documents apportent le nécessaire : l'activité peut être faite après avoir traité tout le cours du chapitre ou avant.

La question **1** consiste à collecter, sur des sites encyclopédiques comme Wikipédia, les données astronomiques de satellites de planètes ou du Soleil, ou des données satellitaires sur [www.n2yo.com](http://www.n2yo.com).

Le traitement réalisé est classique : un tracé pour tester l'expression (donnée) de la troisième loi de Kepler.

L'utilisation de la programmation Python permet de traiter les données automatiquement et de réaliser une régression linéaire.

L'objectif est la détermination de la masse de l'astre attracteur.

La question **2** implique des **modifications du programme** pour calculer les grandeurs à placer en abscisses et en ordonnées, puis pour calculer la masse de l'astre attracteur à partir des données de la régression linéaire.

La question **3** permet de réaliser un tracé en échelle logarithmique, plus adaptée que les échelles linéaires aux grandeurs qui connaissent des grandes variations, comme c'est le cas pour  $T^2$  et  $a^3$ . Seule la modification de `tracelog` de `False` à `True` est nécessaire, le reste est automatique.

## Programme à compléter

```

1 from pylab import *
2 from math import *
3 from scipy.stats import linregress
4 import numpy as np
5 import numpy.polynomial.polynomial as nppol
6
7 print("")
8 print("*****")
9 print("*       Tester la troisieme loi de Kepler       *")
10 print("*               et peser un astre               *")
11 print("*****Hatier 2020*")
12 print("")
13
14 #####
15 ### DONNEES A MODIFIER (Question 1a) ###
16 #####
17
18 ### Nom de l'astre attracteur
19 nomastre="..."
20 ### Masse tabulee de l'astre attracteur en kg
21 M=...
22
23 ### Liste des noms des satellites (entre "")
24 satellites=["...", "...", "...", "...", "...", "..."]
25 ### Liste des demi-grands axes en km
26 a=[...]
27 ### Liste des periodes orbitales en j
28 T=[...]
29
30 ### Donnees a ne pas modifier
31 ### Constante de la gravitation universelle en USI
32 G=6.67e-11
33 ### Initialisation des grandeurs et des tableaux
34 x=np.zeros(len(satellites))
35 y=np.zeros(len(satellites))
36
37 #####
38 ### A MODIFIER (Question 3) ###
39 #####
40 tracelog=False
41
42 #####
43 ### A MODIFIER (Question 2a) ###
44 #####
45 ### Liste des T carres et des a cubes en USI
46 for i in range(len(a)):
47     x[i]=float(...)
48     y[i]=...

```

### Modules

Le module `math` est nécessaire pour les fonctions trigonométriques et la valeur de  $\pi$ . Le module `numpy` permet des opérations sur les listes. Il a été choisi d'utiliser `pylab` pour ses fonctionnalités de tracés. Pour les régressions linéaires, il faut également importer certaines autres fonctions ou modules.

### Données à modifier

Question 1 : le nom de l'astre, sa masse (issue par exemple de Wikipédia), les noms et paramètres orbitaux (demi-grands axes et périodes) de ses principaux satellites. Bien noter que ces valeurs ne sont pas forcément en unités SI et qu'il faut prévoir de les convertir.

### Données à ne pas modifier

La constante de gravitation universelle, l'initialisation des valeurs.

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

### Choix du mode d'affichage

Question 3 : par défaut, `tracelog` est à `False`, et le tracé effectué est  $T^2$  en fonction de  $a^3$ . Pour tracer  $\ln(T)$  en fonction de  $\ln(a)$ , on le règle à `True` et il n'y a rien d'autre à faire.

### Calculs de $T^2$ et $a^3$

Question 2a : il faut en profiter pour faire les conversions en mètres (pour  $a$ ) et en secondes (pour  $T$ ). Si aucune valeur à virgule n'est entrée, c'est aussi l'occasion de déclarer que ces grandeurs sont décimales, avec la fonction `float()`.

```

49
50 ### Ne pas modifier
51 ### Regression lineaire (fonction lineaire)
52 reg=nppol.polyfit(x,y,[1])
53
54 if tracelog==True:
55     for i in range(len(a)):
56         x[i]=log(a[i]*1000)
57         y[i]=log(T[i]*86400)
58 ### Regression lineaire (fonction affine)
59     reg=nppol.polyfit(x,y,[0,1])
60
61
62 m=reg[1]
63 p=reg[0]
64
65 #####
66 ### A MODIFIER (Question 2b) ###
67 #####
68 Mcalc=...
69
70 typereg="Trace de T^2 en fonction de a^3"
71 if tracelog==True:
72     Mcalc=4*pi**2/(G*e**(2*p))
73     typereg="Trace de ln(T) en fonction de ln(a)"
74
75 print("**** Astre etudie:",nomastre,"****")
76 print("****",typereg,"****")
77 print("Coefficient directeur de la regression")
78 print("m=",format(m,"4.3E"),"USI")
79 if tracelog==True:
80     print("Ordonnee a l'origine de la regression")
81     print("p=",format(p,"4.3E"),"USI")
82 print("")
83 print("Masse tabulee de l'astre attracteur")
84 print("M=",format(M,"4.3E"),"kg")
85 print("Masse calculee de l'astre attracteur")
86 print("M=",format(Mcalc,"4.3E"),"kg")
87
88 xmin=0
89 xmax=1.05*max(x)
90 if tracelog==True:
91     xmin=0.95*min(x)
92
93 xmodele=[xmin,xmax]
94 ymodele=[m*xmin+p,m*xmax+p]
95
96 xlabel("a^3 (m^3)")
97 ylabel("T^2 (s^2)")
98
99 if tracelog==True:
100     xlabel("ln(a) (USI)")
101     ylabel("ln(T) (USI)")
102 plot(x,y,"+")
103 for i in range(len(a)):
104     annotate(satellites[i],(x[i],y[i]))
105 plot(xmodele,ymodele)
106 grid(True)
107 show()

```

### Régression linéaire

Question 2b : la fonction `nppol.polyfit` fournit la régression linéaire du tracé de  $y$  en fonction de  $x$ . Si `tracelog` est `False`, alors le tracé est linéaire, de coefficient directeur  $m$ . Si `tracelog` est `True`, alors le tracé est affine, de coefficient directeur  $m$  et d'ordonnée à l'origine  $p$ .

### Calcul de la masse

Question 2b : l'expression de la masse de l'astre fait intervenir le coefficient directeur  $m$  de la régression linéaire si `tracelog` est `False`. Seule cette partie est à modifier. Si `tracelog` est `True`, elle fait intervenir l'ordonnée à l'origine  $p$ .

### Affichage des résultats et tracés

Les grandeurs calculées sont affichées en notation scientifique. Les points sont placés sur un graphique et la droite de régression est tracée.