3eme-loi-de-kepler-3-eleve

June 15, 2023

La 3ème Loi de Képler : 3 - Vérification de la 3ème Loi de Képler

Le but de cette activité est de vérifier la troisième loi de Kepler.

[1]:
```
# -*- coding: utf-8-*-

# Importation des bibliothèques
import numpy as np
import matplotlib.pyplot as plt
import csv
import statistics as stat
```

[2]:
```
def positions(fichier):
    valeurs = []
    Mesures = open(fichier,encoding="utf8", errors='ignore')
    csv_astre = csv.reader(Mesures,delimiter=";")      # On ouvre le fichier csv
    for row in csv_astre:      # On remplit le tableau à partir du fichier csv
        valeurs.append(row)
    valeurs = [list(map(float, x)) for x in valeurs]     # On transforme les␣
 ↪valeurs en flottant (en réels)
    return valeurs
```

[3]:
```
def caracteristiques(fichier):
    coordonnee = np.transpose(positions(fichier))
    aphelie, perihelie = max(coordonnee[3]), min(coordonnee[3]) # coordonnee[3]␣
 ↪est la liste des distances
    a = (aphelie + perihelie) / 2 * 149597870700
    date_min = list(np.where(coordonnee[3] == perihelie))[0][0]
    date_max = list(np.where(coordonnee[3] == aphelie))[0][0]
    T = np.abs(float(date_max-date_min))*2 * positions(fichier)[0][0]* 86400
    return a,T
```

[4]:
```
def caracteristiques_moyennes(fichier):
    Planete = []
    DemiGrand_axe = []
    Periode = []
    ligne = positions(fichier)
    tranche = int(len(ligne)/10)
```

```
    for i in range(10):
        Planete.append([])
        for j in range(i*tranche,(i+1)*tranche):
            Planete[i].append(ligne[j])
        aphelie, perihelie = max(np.transpose(Planete[i])[3]), min(np.
 ↪transpose(Planete[i])[3]) # coordonnee[3] est la liste des distances
        a = (aphelie + perihelie) / 2 * 149597870700
        date_min = list(np.where(np.transpose(Planete[i])[3] ==␣
 ↪perihelie))[0][0]
        date_max = list(np.where(np.transpose(Planete[i])[3] == aphelie))[0][0]
        T = np.abs(float(date_max-date_min))*2 * positions(fichier)[0][0]* 86400
        DemiGrand_axe.append(a)
        Periode.append(T)
    return DemiGrand_axe, Periode
```

[5]:
```
def estimation(liste):
    moyenne = stat.mean(liste)
    return moyenne
```

I. Test de la 3ème loi de Képler

Nous allons estimer, pour toutes les planètes, la valeur de la période de révolution autour du Soleil ainsi que la valeur du demi-grand axe.

Ces valeurs seront regoupés dans 2 fichiers : **periode, demiga**.

[6]:
```
Planete = ['Mercure_H.csv','Venus_H.csv','Terre_H.csv','Mars_H.csv','Jupiter_H.
 ↪csv','Saturne_H.csv','Uranus_H.csv','Neptune_H.csv']
periode, demiga = [],[]
for planete in Planete:
    demiga.append(estimation(caracteristiques_moyennes(planete)[0]))
    periode.append(estimation(caracteristiques_moyennes(planete)[1]))
```

Nous allons tracer la courbe de la distribution de la période en fonction de la valeur du demi-grand axe pour chaque planète du système solaire à l'aide d'axe logarithmiques.

Question 11 : Compléter la seconde ligne du programme ci-dessous (remplacez les .....).

[7]:
```
plt.clf()
x = demiga
y = periode
plt.scatter(.....,......,color='blue',marker = '+')
plt . title (" Evolution de T en fonction de a (axes logarithmiques) ")
plt . xlabel (" demi-grand axe (a) ")
plt . ylabel (" Periode (T) ")
plt.xscale('log')
plt.yscale('log')
plt.show()
```

Question 12 : A partir de la 3ème loi de Képler, établie pour les orbites circulaires, montrer que T = C.a3/2 avec C = 5,45.10-10SI

Question 13 : Compléter le code afin de tracer le T = f(a) à l'aide de la 3ème loi de Képler.

[10]:
```
plt.clf()
C = .....
x = demiga
y = periode
z = [C*a**1.5 for a in x]
plt.scatter(x,y,color='blue',marker = '+')
plt.plot(.....,.....,color='red')
plt . title (" Evolution de T en fonction de a (axes logarithmiques) ")
plt . xlabel (" demi-grand axe (a) ")
plt . ylabel (" Periode (T) ")
plt.xscale('log')
plt.yscale('log')
plt.show()
```

Question 14 : Conclure

Les planètes du système solaire suivent parfaitement la 3ème Loi de Képler

II. La 3ème loi de Kepler reste-t-elle valable pour les trajectoires non circulaires?

En cours, la 3ème loi de Képler a été établie pour les mouvement circulaires.

Les 8 planètes du système solaire ont une orbite elliptique de faible excentricité (c'est à dire une orbite quasi-circulaire).

Qu'en est-il des objets de plus forte excentricité (proche de 1) ?

Nous allons nous intéresser à la comète de Halley (excentricité 0,967; fichier 'Halley_H.csv') et l'astéroïde Halloween (excentricité 0,860;fichier 'Halloween_H.csv').

Question 15 : Compéter le code suivant de façon à faire apparaître l'astéroïde Halloween et la comète de Halley sur la figure.

[ ]:
```
plt.clf()
C = 5.45*10**(-10)
x = demiga
y = periode
z = [C*a**1.5 for a in x]
m = estimation(caracteristiques_moyennes('Halley_H.csv')[0])
n = estimation(caracteristiques_moyennes('Halley_H.csv')[1])
o = estimation(caracteristiques_moyennes(.....)[0])
p = estimation(caracteristiques_moyennes(.....)[1])
plt.scatter(m,n,color='orange',marker = '*')
plt.scatter(o,p,color='green',marker = '*')
plt.scatter(x,y,color='blue',marker = '+')
plt.plot(x,z,color='red')
plt . title (" Evolution de T en fonction de a (axes logarithmiques) ")
```

```
plt . xlabel (" demi-grand axe (a) ")
plt . ylabel (" Periode (T) ")
plt.xscale('log')
plt.yscale('log')
plt.show()
```

Question 16 : Conclure

[ ]:

```

```