TP 16 - Lois de Kepler

Introduction

Au cours de ce TP, on se propose d'adopter la démarche scientifique de Johannes Kepler (1571 - 1630) afin de retrouver ses trois lois bien connues.

En son temps, Kepler s'est basé sur les observations du mouvement des planètes faites par Tycho Brahe (1546 - 1601) pour calculer leur orbite. Il découvre ainsi en 1609 deux premières lois régissant le mouvement des planètes : **la loi des orbites** et **la loi des aires**. Il lui faudra travailler 9 années de plus pour qu'il découvre la dernière loi : **la loi des périodes**.

On a recuilli sur sur le site IMCCE les éphémérides de position des différentes planètes du système solaires. À partir de ces données, nous allons vérifier la deuxième et la troisième lois de Kepler.

Troisième loi de Kepler

Question

1. **Donner** la troisième loi de Kepler.

Manipulation

- **Choisir** une planète.

In [ ]:
```
#CELLULE NON MODIFIABLE

#Bibliothèques
from IPython.display import display, Latex
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
import csv
```

In [ ]:
```
#CELLULE MODIFIABLE 1

#Choix de la planète
planete = #À compléter
```

In [ ]:
```
#CELLULE NON MODIFIABLE

#Ouverture du fichier de données concernant la planète étudié
with open(planete+'.csv') as source:
    donnees = csv.DictReader(source, delimiter=";")
    descripteurs = donnees.fieldnames
    donnees = list(donnees)
```

```
#Obtention des instants et des positions associées en coordonnées cartésiennes
dates = [donnee["Date (undefined)"] for donnee in donnees]
dates = [date[:10] for date in dates] #Suppression des minutes et des heures
pas_temps = np.abs(int(dates[28][-2:])-int(dates[29][-2:])) #Durée entre le deuxième et le premier instant
derniere_date = len(dates)*pas_temps

instant_t = np.arange(0,derniere_date,pas_temps) #Instants des positions en jour
x = np.array([float(donnee["px (au)"]) for donnee in donnees]) #Position selon la coordonnée x en u.a. (unité astro
y = np.array([float(donnee["py (au)"]) for donnee in donnees]) #Position selon la coordonnée y en u.a.
```

Manipulations

- **Utiliser** ce lien pour trouver les fonctions python permettant **d'afficher** dans la cellule modifiable 2 la trajectoire de la planète.
- Toujours dans la cellule modifiable 2, **déduire** de la trajectoire dessinée, le demi-grand axe a de la trajectoire.
- Toujours dans dans la cellule modifiable 2, **déduire** de la trajectoire dessinée, la période de la trajectoire T.
- **Entrer** les valeurs de a et de T dans la cellule modifiable 3 et **recommencer** les manipulations pour les autres planètes.
- **Illustrer** la troisième loi de Kepler à l'aide d'un graphique permettant de lier linéairement des grandeurs impliquant a et T pour toutes les planètes.
- Une fonction d'ajustement donne le coefficient qui lie les grandeurs impliquant a et T. **Noter** ce coefficient.

In [ ]:
```
#CELLULE MODIFIABLE 2

plt.figure(1)
plt.title('Trajectoire de '+planete)

 #À compléter

plt.grid()
plt.xlabel(r'$x$ (u.a.)')
plt.ylabel(r'$y$ (u.a.)')
plt.axis("equal")

rho = #À compléter coordonnée radiale de la base cylindro-polaire
grand_axe =  #À compléter à partir de rho

#1 unité astronomique (la distance Terre-Soleil) est égale à 149 597 870 700 m
demi_grande_axe = 149597870700*grand_axe*0.5
display(Latex(r'Le demi-grand axe $a$ de la trajectoire elliptique est égale %2.3e m.' % ( demi_grande_axe )))

#À compléter à partir de instant_t et rho
instant_1 =  #Instant correspondant au passage de la planète à la coordonnée radiale maximale
instant_2 =  #Instant correspondant au passage de la planète à la coordonnée radiale minimale

periode = 2.*np.abs(instant_2 - instant_1)
display(Latex(r'La période $T$ de la trajectoire elliptique est égale %2.3e jour.' % ( periode )))
```

In [ ]:
```
#CELLULE NON MODIFIABLE

def fonction_d_ajustement(x, a, b):
    return a*x+b
```

In [ ]:
```
#CELLULE MODIFIABLE 3

#Demi-grand axe des trajectoires des planètes
a_planetes = np.array([ ]) #À compléter en m

#Périodes des planètes
periode_planetes = np.array([]) #À compléter en jour


periode_planetes = periode_planetes*3600*24 #seconde

plt.figure(2)
plt.title('Comparaision des demi-grands axes et des périodes des planètes'+"\n")
plt.plot(a_planetes**3,periode_planetes**2,'x')
plt.grid()
plt.xlabel(r'$a^3$')
plt.ylabel(r'$T^2$')
```

```
#Ajustement des points de mesure
popt,_ = curve_fit(fonction_d_ajustement, a_planetes**3, periode_planetes**2)
coefficient = popt[0] #Valeurs des coefficients de l'ajustement et leur incertitude associée
ordonnee_ajustee = fonction_d_ajustement(a_planetes**3,*popt)

plt.plot(a_planetes**3,ordonnee_ajustee,'r')

display(Latex(r'Le coefficient d`ajustement est égale %2.3e SI.' % ( coefficient )))
```

Question

2. **Déterminer** à quoi correspond le coefficent obtenu en considérant que la masse des planètes est négligeables devant celle du Soleil.
3. **Comparer** ce coefficient à sa valeur théorique. *Données :* $G = 6,6743\times10^{-11}$ m^3 $kg^{-1}s^{-2}$ et $M_S =1,98892\times 10^{30}$ kg.

Deuxième loi de Kepler

Question

2. **Donner** la deuxième loi de Kepler.

Pour calculer l'aire S d'un triangle dont les longueurs des côtés sont a, b et c, et le demi-périmètre est $$ p = \frac{a+b+c}{2} $$ on peut utiliser la formule de Héron $$ S= \sqrt{p\left(p-a \right)\left(p-b \right)\left(p-c \right)}. $$ On rappelle que la distance d entre deux points M et N de coordonnées (x_M,y_M) et (x_N,y_N) est $$ d = \sqrt{\left(x_N-x_M \right)^2 + \left(y_N-y_M \right)^2}. $$

Manipulations

- **Calculer et présenter graphiquement** la variation de la vitesse aérolaire d'une planète de votre choix.

In [ ]:
```
#CELLULE MODIFIABLE 4

instant_t_areolaire = instant_t[0:-1]
vitesse_areolaire = np.zeros(len(instant_t)-1)

#À compléter
for k in range(len(instant_t)-1):
    a =  #distance entre le foyer de l'ellipse et le point M
    b =  #distance entre le foyer de l'ellipse et le point N
    c =  #distance entre les points M et N
    p = 0.5*(a+b+c)
    aire = np.sqrt(p*(p-a)*(p-b)*(p-c) )
    vitesse_areolaire[k] = aire/pas_temps

plt.figure(3)
plt.title(r'Variation de la vitesse aréolaire $\mathcal{A}$ de '+planete+"\n")
plt.plot(instant_t_areolaire,vitesse_areolaire)
plt.xlabel(r'$t$ (jour)')
plt.ylabel(r'$\mathcal{A}$ (m$^{2}$ jour$^{-1}$)')
plt.grid()

variation_max_relative = 100*(np.max(vitesse_areolaire)-np.min(vitesse_areolaire))/\
(np.max(vitesse_areolaire)+ np.min(vitesse_areolaire))

display(Latex(r'La variation relative maximale de $\mathcal{A}$ est de %2.3e %%.' % ( variation_max_relative )))
```