CN7

Trajectoires d'un point matériel soumis à un champ de force centrale conservatif (trajectoires des satellites)

Compétence attendue par le programme 2021 : « à l'aide d'un langage de programmation, obtenir des trajectoires d'un point matériel soumis à un champ de force centrale conservatif ».

1. Trajectoire d'une planète dans le système solaire

On va étudier le mouvement d'une planète dans le système solaire, le Soleil est considéré fixe au point O (voir *Fig. 018*). On se place dans le référentiel héliocentrique. Les planètes subissent une force centrale de la part du soleil. On admet que les planètes décrivent autour du Soleil une trajectoire elliptique dont le Soleil est un des foyers.

En coordonnées polaires, les coniques sont décrites par une équation de la forme :

$$r(\theta) = \frac{p}{1 + e\cos\theta}$$

où e est l'excentricité de la conique et $p > 0$ est le paramètre de la conique et décrit qualitativement sa taille. Pour une ellipse, l'excentricité vérifie : $0 < e < 1$.

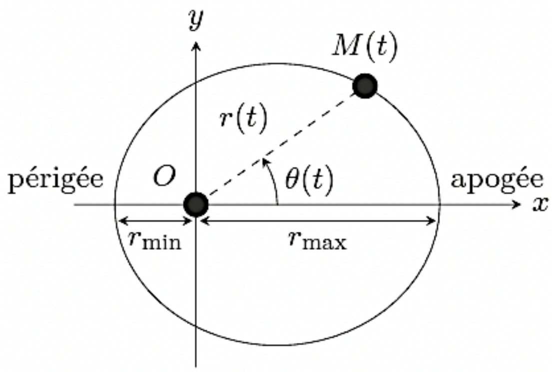

Fig. 018 Trajectoire elliptique d'une planète

Remarque importante :

on a vu dans le cours que l'expression d'une conique était : $r(\theta) = \frac{p}{1+e\cos\theta}$, avec cette expression on constate que $r_{min} = r(0) = \frac{p}{1+e}$ et que $r_{max} = r(\pi) = \frac{p}{1-e}$.

Vous vous demandez peut-être pourquoi il y a un signe moins dans cet exercice $r(\theta) = \frac{p}{1-e\cos\theta}$. avec cette expression on constate que $r_{min} = r(\pi) = \frac{p}{1+e}$ et que $r_{max} = r(0) = \frac{p}{1-e}$. et sur le schéma proposé ci-dessus, on constate bien que $r_{min} = r(\pi)$ et que $r_{max} = r(0)$ ce qui explique qu'on note la conique $r(\theta) = \frac{p}{1-e\cos\theta}$ dans l'exercice proposé.

2. Tracé d'une trajectoire en coordonnées polaires

On se propose de tracer l'ellipse d'équation :

$$r(\theta) = \frac{1,5}{1 - 0,5\cos\theta}$$

Pour cela on importe les bibliothèques nécessaires puis on définit un array pour les valeurs de θ allant de 0 à 2π et l'array de valeurs de r correspondant :

```
import matplotlib.pyplot as plt
import numpy as np
theta = np.linspace(0, 2 * np.pi, 100)#100 valeurs
r = 1.5 / (1 - 0.5*np.cos(theta))#rayon de
    courbure
```

Pour le tracé de l'ellipse, il faut mettre les axes en coordonnées polaires, pour cela on doit ajouter l'option `projection='polar'` :

```
fig = plt.figure()#création d'une figure
ax = fig.add_subplot(111, projection='polar')#
    passage en polaires
```

On peut ensuite tracer et indiquer des valeurs de θ et de r sur le graphique :

```
ax.plot(theta,r)#tracé de l'ellipse
ax.set_xticks(np.arange(0,2.0*np.pi,np.pi/6.0))#
    valeurs de theta
ax.set_yticks(np.arange(0,4,1.0))#valeurs de r
plt.show()
```

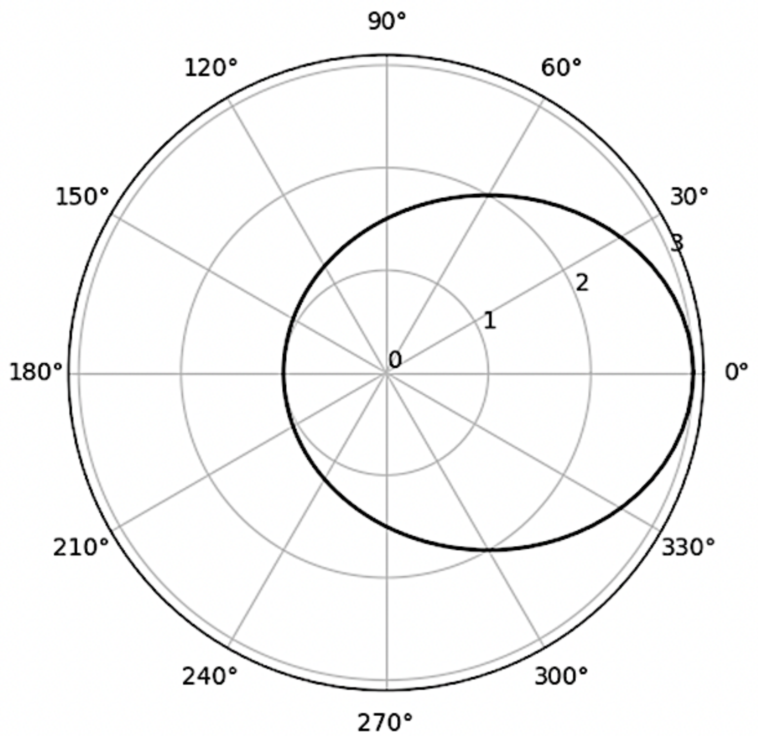

Figure 2 : Tracé d'une trajectoire d'une planète en coordonnées polaires

3. Tracé de plusieurs trajectoires en coordonnées polaires

Ce que vous devez faire : compléter les programmes dans les lignes présentant le symbole ➡ . Faire tourner le programme. Vous essayerez d'utiliser les représentations +++, --- et _____ pour représenter les 3 ellipses.

On souhaite visualiser l'influence des paramètres p et e. Aussi, on va tracer les trajectoires correspondant aux équations :

$$r_1(\theta) = \frac{1,5}{1 - 0,5\cos\theta} \qquad r_2(\theta) = \frac{2}{1 - 0,5\cos\theta} \qquad r_3(\theta) = \frac{1,5}{1 - 0,7\cos\theta}$$

Le principe des tracés est tout à fait identique au précédent et le code donne :

```
theta = np.linspace(0, 2 * np.pi, 100)#100 valeurs
r1 =                                 #ellipse 1
r2 =                               #ellipse 2
r3 =                                 #ellipse 3
fig = plt.figure()
ax = fig.add_subplot(111, projection='      ')
ax.plot(theta,r1,'-',label='r1')#tracé ellipse 1
                                    #tracé ellipse 2
                                   #tracé ellipse 3
ax.set_xticks(                                  )
ax.set_yticks(                     )valeurs de r
ax.legend()#affichage d'une légende
plt.
```