



Lois de Kepler

Ce qu'il faut savoir et savoir faire

Exploiter, à l'aide d'un langage de programmation, des données astronomiques ou satellitaires pour tester les deuxième et troisième lois de Kepler.

Code Capytale : 55bc-10265530

I. Deuxième loi de Kepler

Dans cette partie, nous mettons en évidence la deuxième loi de Kepler. Elle stipule que le **segment de droite qui relie le soleil à une planète parcourt des aires égales dans des temps égaux**.

Dans un premier temps, on récupère les positions héliocentriques de Mars à partir de données fournies par le portail d'éphémérides de l'IMCCE : <https://ssp.imcce.fr/forms/ephemeris>.

- Choisir le corps du système solaire : *Mars*
- Dans Epoque, modifier le nombre de dates à calculer et le pas temporel. *Pour Mars : 1 calcul par jour pendant 700 jours permet d'avoir une précision correcte vu qu'une année martienne dure environ 686,98 jours solaires.*
- Pour les autres champs à renseigner, on choisit :
 - Système de coordonnées : *héliocentre*
 - Plan de référence : *équiptique*
 - Type d'éphémérides : *astrométrique J2000*
 - Coordonnées : *sphériques*
- Cliquer sur « Calculer ».

On obtient un tableau des positions de l'astre, que l'on peut exporter dans un fichier au format **.csv**. **Ce fichier comporte une ligne par date calculée, et plusieurs paramètres en colonnes, dont les coordonnées de l'astre**. Les distances sont données en unité astronomique ($1 \text{ ua} = 149,6 \cdot 10^6 \text{ km}$) et les angles en degrés.

Pour traiter des données, nous allons utiliser la bibliothèque **Python Pandas**.

La fonction `read_csv()`, permet de lire des enregistrements de fichiers **.csv** dans un DataFrame.

- Enregistrer le fichier exporté sous format **.csv**.

Le mouvement de Mars étant quasiment dans le plan de l'équiptique, on assimile la grandeur D_{obs} à la coordonnée polaire r et la longitude à la coordonnée polaire θ .

Q1. Taper le code suivant afin de récupérer une **liste r** des valeurs de r et une **liste theta** des valeurs de θ . (le fichier mars.csv est disponible dans l'activité dans Capytale)

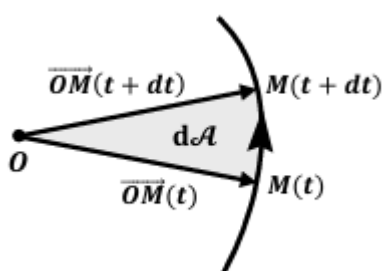
```
import pandas as pd # import de données et gestion en tableau

df = pd.read_csv('mars.csv', delimiter=';')
theta = df["Longitude(deg)"]
r=df["Dobs (au)"]
```

Q2. Calculer pour chaque point ses coordonnées cartésiennes (x, y) et représenter sur un graphe la trajectoire de Mars. A-t-on l’allure attendue ?

Après avoir tracé la trajectoire de Mars, on veut calculer les aires de trois secteurs angulaires balayés par le vecteur position sur un intervalle de temps de même durée mais sur des parties différentes de la trajectoire.

On peut approximer l’aire balayée dA par le vecteur position, pendant une durée dt par :



$$dA \approx \frac{1}{2} \|\overline{OM}(t) \wedge \overline{OM}(t + dt)\|$$

$$= \frac{1}{2} \|\overline{OM}(t)\| \|\overline{OM}(t + dt)\| \left| \sin(\overline{OM}(t), \overline{OM}(t + dt)) \right|$$

$$dA = \frac{1}{2} r(t)r(t + dt) |\sin d\theta|$$

- Q3.** Implémenter un code permettant de calculer l’aire balayée sur 3 intervalles de temps égaux (100 jours par exemple), pris à différents moments de la trajectoire et vérifier qu’elle reste constante.
- Q4.** En déduire la vitesse aréolaire et la constante des aires pour la trajectoire de Mars autour du Soleil.
- Q5.** Option : tracer les trois secteurs angulaires étudiés. Pour cela, on pourra superposer à la figure précédente les différents vecteurs positions correspondant aux dates comprises dans les intervalles de temps étudiés.

II. Troisième loi de Kepler

On donne pour les différentes planètes du système solaire le demi-grand -axe de leur trajectoire et la période de leur mouvement.

Planètes	Demi grand axe $a(ua)$	Période
Mercure	0.387	87.97 j
Vénus	0.723	224.7 j
Terre	1	365.26 j
Mars	1.524	1 an 322 j
Jupiter	5.203	11 ans 315 j
Saturne	9.555	29 ans 167 j
Uranus	19.22	84 ans 27 j
Neptune	30.10	164 ans 280 j

Q6. Faire une régression linéaire permettant de tester la troisième loi de Kepler et d’en déduire la masse du Soleil.