# TP21 – Astronomie

## Objectifs

→ Procéder à l'évaluation d'une incertitude-type par une approche statistique.
→ **Exploiter, à l'aide d'un langage de programmation, des données astronomiques ou satellitaires pour tester les deuxième et troisième lois de Kepler.**

## Données

| | |
|---|---|
| constante gravitationnelle : | $G = 6{,}67 \times 10^{-11}\,\mathrm{N \cdot m^2 \cdot kg^{-2}}$ |
| distance Terre – Soleil : | $1\,\mathrm{ua} = 1{,}496 \times 10^{11}\,\mathrm{m}$ |
| masse du Soleil : | $M_\odot = 1{,}988 \times 10^{30}\,\mathrm{kg}$ |
| parsec : | $1\,\mathrm{pc} = 3{,}09 \times 10^{16}\,\mathrm{m} = 3{,}26$ années lumières |

## Deuxième loi de Kepler : Système solaire

On se propose de vérifier la deuxième loi de Kepler, ou loi des aires, en vérifiant la conservation de la constante des aires à partir des positions de quelques objets du Système solaire.

1. Récupérer sur cahier-de-prepa l'archive `tp21.zip` dont on extraira le contenu dans le répertoire de travail. Le programme `tp21-deuxieme_loi.py` permet de lire et représenter les données du fichier `data/earth.txt` contenant les positions de la Terre, relevées chaque jour pendant 1000 jours.

   APP À quoi correspondent les données des colonnes 1 à 4 et 8 à 10 du fichier `data/earth.txt` utilisées pour remplir les tableaux `t`, `x`, `y`, `z`, `vx`, `vy` et `vz`. Indiquer les unités de ces grandeurs.

2. REA VAL On souhaite vérifier la conservation de la norme du moment cinétique, ce qui revient à montrer que $\mathcal{C} = \|\overrightarrow{OM} \wedge \vec{v}\| = \text{cste}$. Exprimer $\mathcal{C}$ en fonction de $x$, $y$, $z$, $v_x$, $v_y$ et $v_z$. Représenter alors graphiquement $\mathcal{C}$ en fonction de $t$. Proposer une explication aux fluctuations observées.

3. REA VAL RCO Indiquer la valeur de la constante des aires $\mathcal{C}_{\text{moy}}$ obtenue, accompagnée de son incertitude-type $u(\mathcal{C})$. Représenter graphiquement l'évolution temporelle de l'écart normalisé $E_n(t)$, défini par :

$$E_n(t) = \frac{\mathcal{C}(t) - \mathcal{C}_{\text{moy}}}{u(\mathcal{C})}.$$

   Conclure quant à validité de la deuxième loi de Kepler dans le cas de la Terre.

L'orbite de la Terre a une excentricité très faible : sa trajectoire est quasi circulaire et le mouvement est presque uniforme. On s'intéresse dorénavant, au choix :

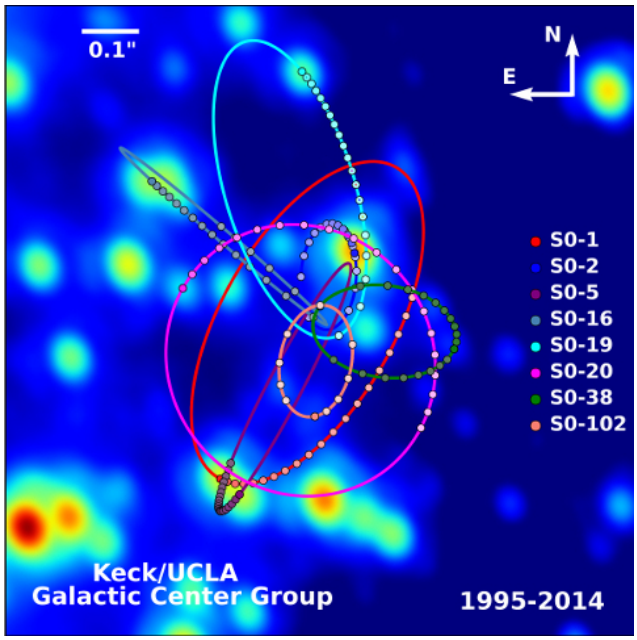
- à la comète 67P/Churyumov-Gerasimenko ayant une période orbitale d'environ 6,5 ans, sur laquelle s'est posé le module Philae de la mission Rosetta en 2014 ;
- à l'astéroïde géocroiseur 1685 Toro ayant une période orbitale d'environ 1,6 ans.

4. ANA REA VAL Vérifier la loi des aires à partir des relevés de position accessibles sur le site de l'Institut de mécanique céleste et de calcul des éphémérides (IMCEE, Doc. 1), pour l'un de ces corps.

## Troisième loi Kepler : Sgr A*

On s'éloigne provisoirement du Système solaire pour se rapprocher du centre de la Voie Lactée, où se trouve Sagittarius A* (Sgr A*), une source intense d'ondes radio. L'étude des trajectoires des étoiles en orbite autour de Sgr A* a permis de déterminer sa masse et de donner une limite haute à son rayon : on en conclu qu'il s'agit d'un trou noir supermassif.

On utilisera les paramètres orbitaux des étoiles répertoriées dans le tableau 3 de l'article Stellar Orbits around the Galactic Center Black Hole Ghez *et al.*, paru en 2005 dans la revue The Astrophysical Journal. La première auteur de l'article est l'une des récipiendaires du prix Nobel de physique en 2020 pour la découverte de ce trou noir.




*Dans l'article, la période est notée $P$. Le demi grand-axe se traduit par semimajor axis.*

APP ANA **5.** En s'appuyant sur la figure 2 de l'article, interpréter les incertitudes associées aux paramètres orbitaux des étoiles listées, en particulier pour S0-1, S0-4 et S0-5.

APP ANA REA VAL RCO **6.** Rappeler la troisième loi de Kepler. À l'aide d'une représentation graphique pertinente, vérifier la troisième loi de Kepler à l'aide des données du tableau 3. On s'appuiera sur un ajustement linéaire.

REA VAL **7.** Exprimer la masse $M$ de Sgr A* en fonction des paramètres orbitaux des étoiles étudiées et de $G$. En déduire une valeur de $M$ exprimée en unité de masse solaire et accompagnée de son incertitude-type, estimée par une méthode statistique. Commenter la valeur obtenue par rapport à celle indiquée par les auteurs dans le tableau 2 ou la conclusion de l'article.

REA **8.** D'autres mesures permettent d'estimer la distance $R_0$ séparant le Système solaire du centre galactique. La valeur déterminée par la collaboration GRAVITY est $R_0 = 8178\,\text{pc}$. En déduire la période de révolution du Système solaire dans le référentiel galactocentrique.

ANA **9.** Dans l'article, la valeur de la masse $M$ est suivie de la notation $[R_0/(8\,\text{kpc})]^3$. Proposer une explication.

## Troisième loi de Kepler : Système solaire

On revient au Système solaire de manière à déterminer la masse du Soleil à partir des relevés de position de l'IMCEE. On souhaite pour cela extraire efficacement la période $T_k$ et le demi grand-axe $a_k$ des orbites de plusieurs corps. Le dossier `data/` contient les fichiers de données associées aux huit planètes du Système solaire, auxquels on pourra rajouter ceux d'autres astres (Pluton, Charon, comètes de Halley, d'Hale-Bopp, astéroïdes Cérès, Vesta, etc.).

On pourra travailler à partir du programme `tp21-troisieme_loi_soleil.py`.

APP REA **10.** Exprimer la distance $r$ des coordonnées cylindriques entre un corps et le soleil en fonction de ses coordonnées $x$, $y$ et $z$ dans le repère cartésien héliocentrique. Représenter graphiquement l'évolution de $r$ en fonction du temps dans le cas de Mercure. Commenter la courbe obtenue et estimer la période de révolution de Mercure.

REA **11.** Exprimer et calculer le demi grand-axe $a$ de sa trajectoire en fonction des valeurs extrémales $r_{\min}$ et $r_{\max}$.

ANA REA **12.** La fonction `theta(x,y,z)` permet d'obtenir la coordonnée $\theta$ des coordonnées cylindriques. Représenter graphiquement $\theta(t)$. Justifier que la pente moyenne de cette courbe permet d'obtenir la période de révolution $T$. La déterminer pour Mercure.

APP ANA REA VAL **13.** Proposer et mettre en œuvre un protocole permettant d'exploiter systématiquement les données de positions des différents fichiers pour vérifier la troisième loi de Kepler, puis déterminer la masse du Soleil et son incertitude-type.

# Documents

## Document 1 – Éphémérides

L'IMCEE met à disposition un service permettant d'obtenir les éphémérides de nombreux objets du Système solaire, c'est-à-dire leurs positions à différentes dates. Pour cela, il faut remplir les différents onglets du formulaire https://ssp.imcce.fr/forms/ephemeris.

- Corps du Système solaire : choisir le corps étudié, soit parmi les astres principaux, soit en effectuant une recherche ;
- Époque : choisir le nombre de points et le pas de temps séparant deux positions successives ;
- Système de coordonnées : choisir héliocentre pour l'origine du repère, écliptique comme plan de référence et cartésiennes ;
- Cliquer sur calculer.

On peut ensuite télécharger les données afin de les exploiter avec Python :

- cliquer sur l'icône ⬇ ;
- choisir Fichier texte (txt) et Jour julien ;
- cliquer sur exporter ;
- copier-coller les données dans un fichier texte que l'on placera dans le répertoire `data/`, situé dans le même répertoire que le programme Python.

## Document 2 – Quelques fonctions utiles

`np.mean(tab)` : moyenne des valeurs contenues dans la liste ou le tableau `tab`.

`np.std(tab, ddof=1)` : écart-type expérimental des valeurs de la liste ou du tableau `tab`.

`np.min(tab)`, `np.max(tab)` : valeurs minimale et maximale de la liste ou du tableau `tab`.

`numpy.polyfit(x, y, 1)` : réalise un ajustement linéaire des données contenues dans les tableaux `x` et `y`. Renvoie les paramètres de la droite d'équation $y = ax + b$ sous la forme d'un tableau : `[a, b]`.

`plt.semilogx`, `plt.semilogy`, `plt.loglog` : remplacent la commande `plt.plot` pour tracer un graphe en échelle logarithmique (abscisse seulement, ordonnée seulement ou les deux).

# Annexes

## Annexe 1 – tp21-deuxieme_loi.py

```python
import numpy as np
import matplotlib.pyplot as plt

data = np.loadtxt("data/earth.txt", skiprows=18, max_rows=1000)
t          = data[:,0] - data[0,0]
x, y, z    = data[:,1], data[:,2], data[:,3]
vx, vy, vz = data[:,7], data[:,8], data[:,9]

# REPRÉSENTATIONS GRAPHIQUES
fig = plt.figure(1)
plt.title("Trajectoire")
plt.xlabel("À compléter")
plt.ylabel("À compléter")
plt.scatter(0, 0)
plt.plot(x, y)
fig.axes[0].set_aspect("equal") # même échelle sur les deux axes
plt.show()
```

## Annexe 2 – tp21-troisieme_loi.py

```python
import numpy as np
import matplotlib.pyplot as plt

a = np.array([]) # À compléter
T = np.array([]) # À compléter

# REPRÉSENTATIONS GRAPHIQUES
plt.figure(1)
plt.plot() # À compléter
plt.show()
```

## Annexe 3 – tp21-troisieme_loi_soleil.py

La base du programme est similaire à `tp21-deuxieme_loi.py`. Il contient en plus la fonction `theta(x,y,z)` qui permettent de déterminer la position angulaire $\theta$ d'un point de coordonnées cartésiennes $x$, $y$ et $z$.

```python
def theta(x, y, z):
    """
    params: coordonnées cartésiennes x, y, z
    return: angle theta des coordonnées cylindriques
    """
    return np.unwrap(np.angle(x+1j*y))
```