

## Fiche de présentation et d'accompagnement

Terminale : Enseignement de Spécialité

Discipline(s) dominante(s) : Mécanique

**Chapitre** : Mouvement dans un champ de gravitation

**Nom de l'activité** : Reproduire les observations de Galilée

pour vérifier la troisième loi de Kepler et peser Jupiter.

TP

Travail en groupe

Durée : 2h

### Programme officiel

Savoir	Savoir-faire
<b>Mouvement dans un champ de gravitation</b> Mouvement des satellites et des planètes. Orbite. Lois de Kepler. Période de révolution.	Établir et exploiter la troisième loi de Kepler dans le cas du mouvement circulaire.  Capacité numérique : Exploiter, à l'aide d'un langage de programmation, des données astronomiques ou satellitaires pour tester les deuxième et troisième lois de Kepler.

### Compétences pouvant être évaluées au cours de l'activité

<input type="checkbox"/> S'approprier	<input checked="" type="checkbox"/> Analyser	<input checked="" type="checkbox"/> Réaliser	<input checked="" type="checkbox"/> Valider	<input type="checkbox"/> Communiquer
---------------------------------------	--	--	---	--------------------------------------

### Organisation de la séance et remarques :

Les élèves réalisent le pointage des positions successives des satellites de Jupiter à l'aide d'un programme (libre de droits) simulant une observation à travers un télescope. Ils peuvent ainsi déterminer des grandeurs caractéristiques de ces satellites.

Ils sont ensuite amenés à modifier un programme Python leur permettant de vérifier la 3ème loi de Kepler et de calculer la masse de Jupiter.

**TP : Reproduire les observations de Galilée pour vérifier  
la troisième loi de Kepler et... peser Jupiter !  
Compétences ciblées : ANA, REA, VAL**

- **Objectifs B.O.** : Mesurer les rayons et les périodes de révolution des quatre principaux satellites de Jupiter. Utiliser Python pour vérifier la troisième loi de Kepler.
- **Problématique** : Déterminer la masse de Jupiter (du Soleil)
- **Intro** : En 1609, l'invention du télescope par **Galilée** permet l'observation d'objets invisibles à l'oeil nu. Galilée découvre que Jupiter est entouré de 4 satellites. Il les observe longuement. Le système de Jupiter était particulièrement important par sa ressemblance avec le système des planètes orbitant autour du Soleil. Son étude a aidé à comprendre les mouvements dans le système solaire. Le système de Jupiter montrait que le modèle héliocentrique du système solaire proposé par Copernic était physiquement possible. Malheureusement, l'Inquisition s'inquiéta de ses découvertes et Galilée fut forcé de se renier.

source : [https://pariscosmo.in2p3.fr/sites/default/files/TP\\_Jupiter\\_JLRobert2.pdf](https://pariscosmo.in2p3.fr/sites/default/files/TP_Jupiter_JLRobert2.pdf)

- **Indice n°1** : Les observations de Galilée... avec le logiciel Jupiter

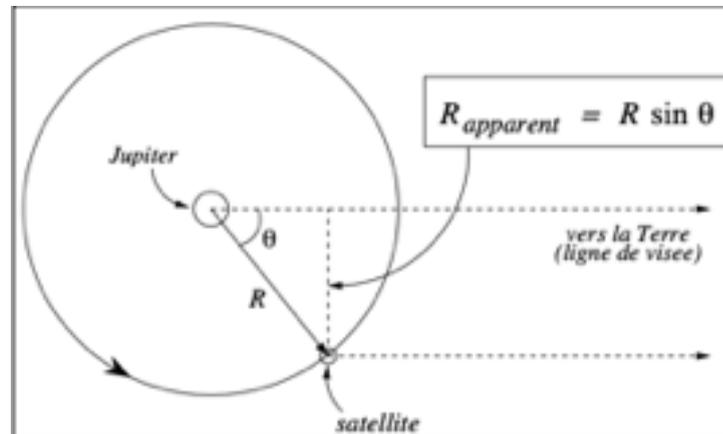
A l'aide du logiciel Jupiter (logiciel libre de droit), on peut observer Jupiter et ses quatre lunes **Io**, **Europe**, **Ganymède** et **Callisto**, par ordre de leur distance, vues à travers un petit télescope. Le logiciel permet de réaliser un pointage de chaque satellite au cours du temps.



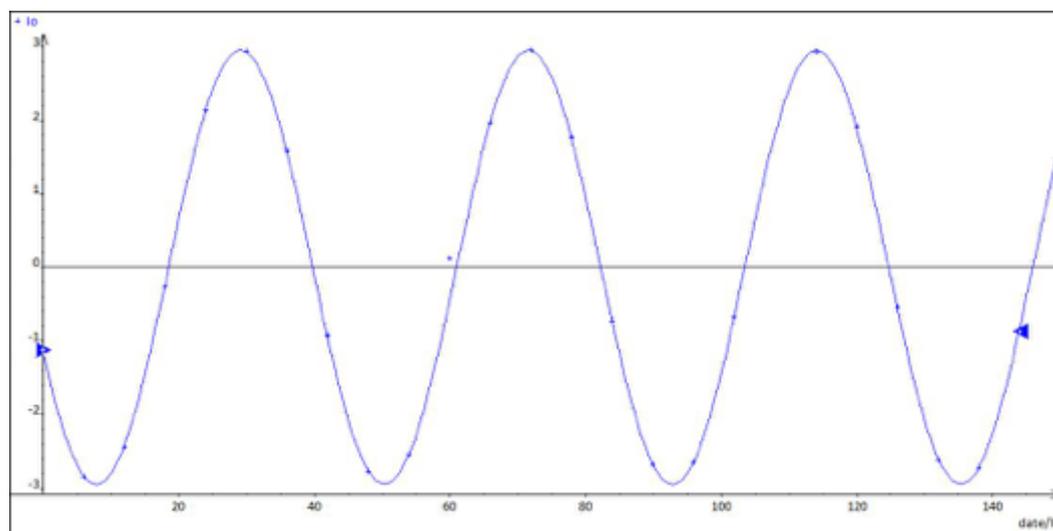
source : <http://www3.gettysburg.edu/~marschal/clea/CLFAhome.html>

- **Indice n°2** : Comment déterminer les caractéristiques d'un satellite (ou lune) ?

Autour de Jupiter, les lunes suivent une orbite à peu près circulaire mais, depuis la Terre, on ne voit que la projection du mouvement dans le plan du ciel : le système est « vu de côté ».



La distance de la lune à la ligne de visée en fonction du temps est donc une courbe sinusoïdale.



En prenant suffisamment de mesures de la position apparente de la lune au cours du temps, on peut tracer une courbe et déterminer le rayon de l'orbite (l'amplitude de la courbe) et la période de l'orbite (la période de la sinusoïde).

On ne peut pas connaître la distance de la lune à Jupiter mais seulement la distance apparente, autrement dit, la distance de la lune à la droite joignant Jupiter à la Terre (ligne de visée), c'est pourquoi le logiciel Jupiter mesure les distances en « Diamètre de Jupiter, J.D. ».

- **Indice n°3** : La troisième loi de Kepler

En 1543 **Nicolaus Copernicus** suppose que les planètes tournent sur des orbites circulaires autour du Soleil. **Tycho Brahé** observe soigneusement l'emplacement des planètes et de 777 étoiles pendant 20 ans en utilisant un sextant et un compas. Ces observations sont utilisées par **Johannes Kepler**, un étudiant de Tycho Brahé pour déduire de manière empirique trois lois mathématiques gouvernant l'orbite d'un objet par rapport à un autre. Pour une lune tournant autour d'une planète, la troisième loi est :

$$\frac{T^2}{a^3} = \frac{4 \times \pi^2}{G \times M}$$

où **M** est la masse de la planète (en kg)

**a** est le demi grand axe de l'orbite elliptique (en km)

**T** est la période (en seconde). La période est le temps nécessaire au satellite pour effectuer un tour complet autour de la planète.

- **Indice n°4** : Effectuer une régression linéaire avec le programme Python

Pour vérifier une loi, on peut, à partir de la représentation graphique des points de mesure, effectuer une régression linéaire. Les lignes de codes fournies peuvent être modifiées si besoin. Elles permettent :

- de représenter graphiquement les points de mesure ;
- d'effectuer une régression linéaire ;
- de calculer la masse du « corps attracteur », dans l'exemple donné : le Soleil.

Le programme Python fourni fonctionne avec le logiciel Edupython. On peut aussi utiliser le site : <https://repl.it> qui permet d'écrire puis de tester un programme en ligne. Attention ! Ce programme doit être modifié avant de l'utiliser pour trouver la masse de Jupiter !

- **Votre travail** :

- Utiliser le logiciel Jupiter puis un tableur-grapheur pour déterminer les caractéristiques (rayons de l'orbite et période) des quatre satellites de Jupiter (**REA**) ;
- Modifier/compléter le code du programme Python afin de vérifier la troisième loi de Kepler (**ANA et NUM**) ;
- Déterminer la masse de Jupiter et la comparer à une valeur de référence (**VAL**)

- **Données utiles : (pour vérifier les mesures et le résultat final)**

Masse de Jupiter :  $1,900 \times 10^{27}$  kg

Satellite	Io	Europe	Ganymède	Callisto
rayon de l'orbite a (en km)	422 000 km	671 000 km	1 070 000 km	1 883 000 km
Période T (en jour)	1,77 j	3,55 j	7,15 j	16,69 j

- **Pointage des lunes de Jupiter :**

Jupiter Satellite Measurements - 04-25-2020 11:12				
	Measurement unit is Jupiter diameters.			
	East is negative (-).			
Date (h)	Io	Europa	Ganymede	Callisto
0	-1.13	-1.90	6.88	
6	-2.85	-3.55	6.05	1.48
12	-2.45	-4.53	4.93	
18	-0.25	-4.68	3.58	-0.98
24	2.13	-3.91	2.09	-2.19
30	2.94	-2.40	0.46	-3.40
36	1.59	-0.43	-1.19	-4.60
42	-0.93	1.64	-2.76	-5.72
48	-2.78	3.37	-4.23	-6.80
54	-2.54	4.43	-5.45	-7.82
60	0.12	4.65	-6.45	-8.85
66	1.97	3.93	-7.13	-9.70
72	2.95	2.49	-7.45	-10.50
78	1.77		-7.43	-11.20
84	-0.74	-1.46	-7.05	-11.80
90	-2.69	-3.22	-6.32	-12.30
96	-2.65	-4.40	-5.30	-12.70
102	-0.67	-4.72	-4.03	-12.95
108	2.93	-2.78	-0.97	-13.15
114	1.92	-0.85	0.68	-13.05
120	-0.54	1.23	2.29	-12.90
126	-2.62	3.07	3.78	-12.55
132	-2.72	4.28	5.08	-12.10
138	-0.87	4.67	6.18	

## • Programme Python : Détermination de la masse du Soleil

```
1 # Programme Python pour vérification 3ème loi de Kepler, créé par GRAC Spé Terminale, le 20/04/2020
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import scipy.stats as sc
5
6 #Affichage des données utiles pour tester le programme
7 print("Vous trouverez ci-dessous les rayons et les périodes des orbites de Vénus, la Terre, Mars et de Jupiter")
8 print("Vénus : rayon orbite = 108 millions de km ; période de révolution = 225 j ")
9 print("Terre : rayon orbite = 150 millions de km ; période de révolution = 365 j ")
10 print("Mars : rayon orbite = 228 millions de km ; période de révolution = 687 j ")
11 print("Jupiter : rayon orbite = 778 millions de km ; période de révolution = 4333 j ")
12
13 #Saisie des rayons ou des demi-grand axes des orbites ainsi que des périodes des planètes
14 a1 = float(input("entrer le rayon de la 1ère planète en millions de km"))
15 print ("le rayon est",a1,"en km")
16 T1 = float(input("entrer la période de la 1ère planète en année"))
17 print ("la période est",T1,"en année")
18
19 a2 = float(input("entrer le rayon de la 2ème planète en millions de km"))
20 print ("le rayon est",a2,"en km")
21 T2 = float(input("entrer la période de la 2ème planète en année"))
22 print ("le rayon est",T2,"en année")
23
24 a3 = float(input("entrer le rayon de la 3ème planète en millions de km"))
25 print ("le rayon est",a3,"en km")
26 T3 = float(input("entrer la période de la 3ème planète en année"))
27 print ("le rayon est",T3,"en année")
28
29 a4 = float(input("entrer le rayon de la 4ème planète en millions de km"))
30 print ("le rayon est",a4,"en km")
31 T4 = float(input("entrer la période de la 4ème planète en année"))
32 print ("le rayon est",T4,"en année")
33
34 #Création des listes pour calcul et affichage
35 a = np.array([a1, a2, a3, a4])
36 T = np.array([T1, T2, T3, T4])
37 G = 6.67e-11
38 #Calcul des constantes a^3 et T^2
39 acube = (a*1e9)**3
40 Tcarre = (T*24*3600)**2
41
42 #Affichage des points de mesure et régression linéaire
43 plt.plot(acube, Tcarre, "+", color = "red")
44 droite = sc.linregress(acube, Tcarre)
45 coeff = droite.slope
46 print ("coeff directeur ", coeff)
47 oorigine=droite.intercept
48 print ("ordonnée à l'origine", oorigine)
49
50 #Calcul de la masse du corps attracteur
51 M = (4*(np.pi)**2)/(G*coeff)
52
53 #Affichage de la droite modélisée
54
55 print ("la masse du Soleil est ", "%0.3e" % M, "kg")
56 Tcarremodelise = coeff*acube+oorigine
57 plt.plot(acube, Tcarremodelise, color = "blue")
58 plt.show()
```