Nom, prénom : CORRECTION

Activité numérique : lois de Kepler et exploitation de données orbitales

Capacité numérique	☺	☹
Exploiter, à l'aide d'un langage de programmation, des données astronomiques ou satellitaires pour tester les deuxième et troisième loi de Kepler		

Objectifs :

- **Exploiter des données orbitales** sur les planètes du système solaire pour déterminer certaines caractéristiques de leur mouvement.
- **Tester les lois de Kepler**.

L'atteinte de ces 2 objectifs passera par l'analyse et l'élaboration de programme en langage Python.

DOCUMENTS et RESSOURCES

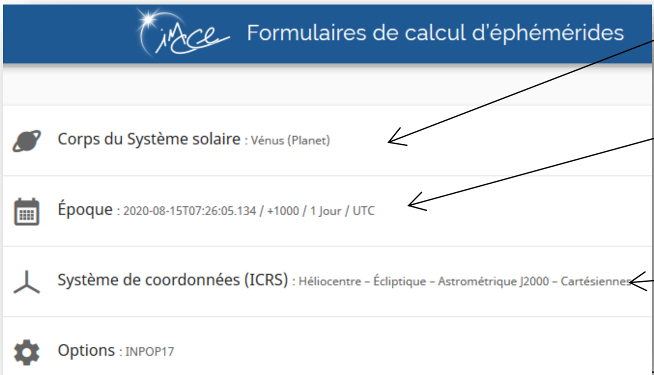
Doc. 1 : lois de Kepler

- La *première loi de Kepler* stipule que les planètes du système solaire décrivent des orbites elliptiques dont le soleil occupe l'un des foyers *(cf. doc. 3)*.
- La *deuxième loi de Kepler* (ou loi des aires) exprime qu'une planète balaie des aires égales en des durées égales.
- La *troisième loi de Kepler* affirme que pour tout astre en orbite autour d'un même astre attracteur, le rapport du carré de la période de révolution sur le cube du demi-grand axe de la trajectoire est une constante.

Doc. 2 : données orbitales de quelques planètes du système solaires sur le site de l'IMCCE

- **Présentation :** Les données orbitales de divers objets célestes sont disponibles sur le site de l'Institut de Mécanique Céleste et de Calcul des Éphémérides. Il est ainsi possible d'obtenir jusqu'à 10 000 valeurs orbitales de différents astres du système solaire. Pour les besoins de cette activité, les données seront à récupérer dans un fichier au format csv. La position et la vitesse de l'astre étudié seront données dans un référentiel d'origine le barycentre du système solaire, le plan de référence étant le plan de l'écliptique. Chaque pointage du fichier comportera 11 informations numériques dont :
  - une date au format *ISO 8601 – AAAA-MM-JJThh :mm :ss.sss* (précision maximale d'une milliseconde)
    → colonne du fichier csv : ***Date***
  - les coordonnées cartésiennes de la position, en unité astronomique
    → colonnes du fichier csv : ***px (au)*** ***py (au)*** ***pz (au)***
  - les coordonnées cartésiennes de la vitesse, en unité astronomique par jour
    → colonnes du fichier csv : ***vx (au/d)*** ***vy(au/d)*** ***vz (au/d)***
- **Protocole de récupération de données orbitales d'un astre dans un fichier csv :**
  1/ Se rendre à l'adresse suivante : https://ssp.imcce.fr/forms/ephemeris

  2/ Faire les sélections demandées dans les menus déroulants



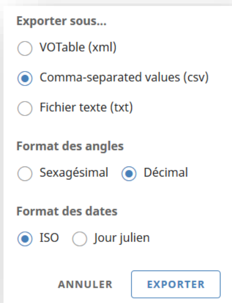
Choix de l'astre étudié Vénus

Temps et échantillonnage (Epoque):
Jour initiale (Date) : aujourd'hui
Echelle de temps : UTC
Nombre de dates : 1000
Fréquence (pas de calcul) : 1 jour

Référentiel (Système de coordonnées) :
Centre du repère : héliocentrique
Plan de référence : plan écliptique
Type d'éphéméride : Astrométrique J2000
Coordonnées : cartésiennes

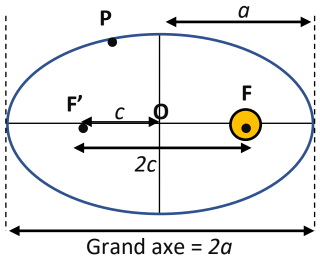
Options : ne pas modifier

3/ Appuyer sur *« Calculer »*
4/ Télécharger les données en cliquant sur puis faire les réglages comme ci-contre :
5/ Cliquer sur « *Exporter* » *puis choisir « Enregistrer le fichier »*.



6/ Ouvrir le dossier de téléchargement, et copier le fichier enregistré.
7/ Coller le fichier dans votre dossier de travail
8/ Renommer le fichier en « *Nomdelaplanète.csv* » *(ne pas oublier l'extension)*

Doc. 3 : caractéristiques d'une ellipse



- Une ellipse est caractérisée par :
  - des points particuliers
    - son **centre O**
    - ses **foyers F et F'** à égales distances du centre O
  - des valeurs particulières
    - **a =** longueur du **demi grand axe**
    - c = distance entre le centre O et un foyer
    - **e = excentricité** = $\frac{c}{a}$: nombre décimal compris entre 0 et 1 qui caractérise l'aplanissement par rapport à un cercle.
- L'ellipse est l'ensemble des points P du plan tels que PF + PF' = 2a

Remarque :
Le cercle est une ellipse particulière où les points O, F et F' sont confondus, l'excentricité e est donc nulle, la longueur du demi-grand axe a est égal au rayon du cercle R.

Doc. 4 : définitions astronomiques

- **Aphélie :** point de la trajectoire d'une planète le plus éloigné du soleil *(la distance entre le soleil et l'aphélie est notée r_a).*
- **Périhélie :** point de la trajectoire d'une planète le plus proche du soleil *(la distance entre le soleil et l'aphélie est notée r_b).*

Les caractéristiques de la trajectoire elliptique de la planète peuvent alors se calculer grâce aux relations suivantes :

$$a = \frac{r_a + r_p}{2} \quad \text{et} \quad e = \frac{r_a - r_p}{r_a + r_p}$$

Doc. 5 : 3 éléments de langage Python

- **3 programmes** en langage Python sont mis à votre disposition pour cette activité, un par loi de Képler.
- **Quelques codes mathématiques** en langage python utiles :
  - Obtenir la valeur minimale d'une liste de valeurs x → *min(x)*
  - Obtenir la valeur maximale d'une liste de valeurs x → *max(x)*
  - Elever un nombre x à la puissance n → *x**n*
  - Obtenir la racine carrée d'un nombre x → *mat.sqrt(x) : nécessite l'import du module « math as mat »*
  - Obtenir la moyenne d'une liste de valeurs *x* → *stat.mean(x) : nécessite l'import du module « statistics as stat »*

VOTRE TRAVAIL

Instructions préliminaires :

- Créer un dossier *« TP Képler »* dans votre ordinateur *qui* accueillera **tous** les fichiers liés à cette activité.

1ère partie : caractériser le mouvement de la planète VENUS du système solaire.

Objectifs :
- *Vérifier la planéité et le caractère elliptique de l'orbite d'une planète.*
- *Déterminer les valeurs de l'excentricité e, du demi-grand axe a de cette ellipse, et la période de révolution T.*

1) Mettre en œuvre la procédure du doc. 2 pour créer un fichier au format *.csv* contenant les données orbitales de la planète étudiée et l'enregistrer dans votre dossier « TP Képler ».

2) Analyse et finalisation du programme 1.

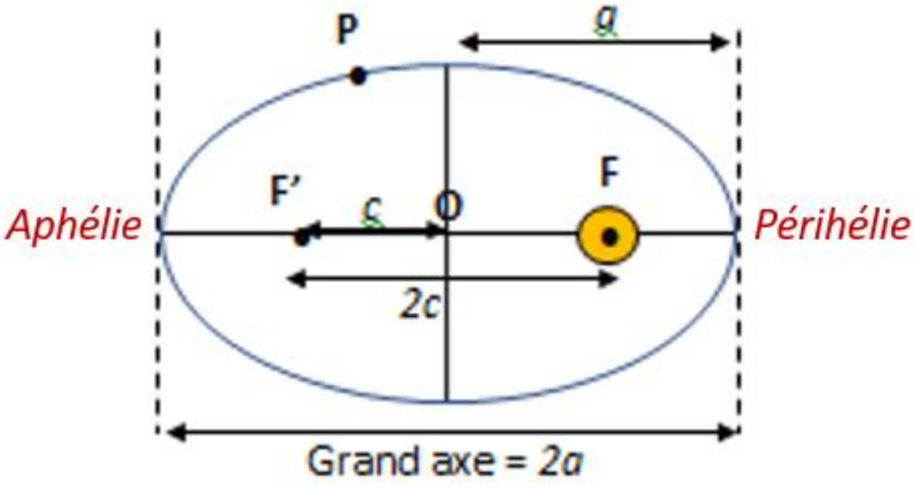
a. Copier le programme 1 ci-dessous dans l'éditeur Python disponible *(Edupython ou autre)*

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
import math as mat

# PARTIE 1 RECUPERATION DONNEES ORBITALES DANS FICHIER CSV
Nom=input('nom de la planète étudiée:')
data = pd.read_csv("venus.csv", sep=";", decimal=".")
X = data['px (au)']
Y = data['py (au)']
Z = data['pz (au)']
print("\nla liste contenant les coordonnées X")
print(X)
print("\nla liste contenant les coordonnées Y")
print(Y)
print("\nla liste contenant les coordonnées Z")
print(Z)

# PARTIE 2 GRAPHE TRAJECTOIRE EN 3D
fig = plt.figure (figsize=(8,8))
ax = fig.add_subplot (111, projection='3d')
ax.plot(X, Y, Z)
plt.title(Nom)
ax.set_xlabel('x(ua)')
ax.set_ylabel('y(ua)')
ax.set_zlabel('z(ua)')
plt.show()

# PARTIE 3 CALCUL d DISTANCE AU SOLEIL
n=(len(X))
d=[]

for i in range(0,n-1) :

   d.append(mat.sqrt(X[i]**2+Y[i]**2+Z[i]**2))

# PARTIE 4 DISTANCES PERIHELIE ET APHELIE
Ra=max(d)
Rp=min(d)
print("\nla distance Ra (en U.A.) de l'aphélie de la trajectoire est ")
print (Ra)
print("\nla distance Rp (en U.A.) du périhélie de la trajectoire est ")
print (Rp)

# PARTIE 5 DEMI GRAND AXE ET EXCENTRICITE
a=(Ra+Rp)/2
e=(Ra-Rp)/(Ra+Rp)
print("\nle demi-grand axe a (en U.A.) de l'ellipse est ")
print (a)
print("\nl'excentricité e (en U.A.) de l'ellipse est ")
print (e)

# PARTIE 6 PERIODE T
## Calcul de la distance D entre la position de la planète et sa position initiale
D=[]
for i in range(0,n-1) :
   D.append(mat.sqrt((X[i]-X[0])**2+(Y[i]-Y[0])**2+(Z[i]-Z[0])**2))
## Calcul de T (Temps mini pour retour à la position initiale)
i=1
while D[i] > 0.01 :
        T=i+1
        i=i+1
print("\nla période de révolution T (en jours) de l'astre est de : ")
print(T)
```

b. Indiquer en surlignant sur le programme, la ligne du programme où la distance entre la planète et le soleil est calculée. Traduire cette expression en python en langage mathématique.

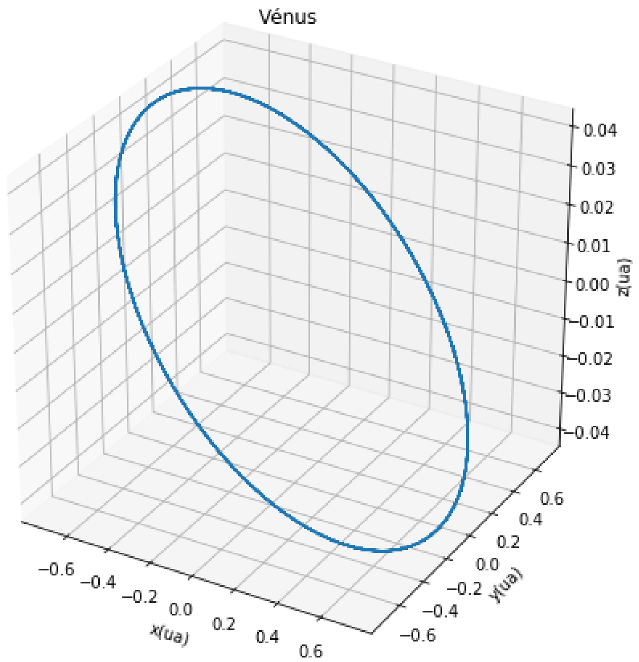
$$d = \sqrt{x^2 + y^2 + z^2}$$

c. Indiquer sur le schéma suivant, la position de l'aphélie et du périhélie en supposant que le soleil est au point F.



d. En vous appuyant sur leur définition, saisir dans le programme les instructions (lignes rouges) qui permettent la détermination des grandeurs R_a et R_p *(écrire les instructions sur le programme)*

e. Compléter les lignes rouges permettant d'obtenir les valeurs de l'excentricité *e* et de la longueur *a* du demi grand axe. *(écrire les instructions sur le programme)* *f.* Exécuter le programme pour obtenir le tracé de la trajectoire de votre planète. *Enregistrer la courbe obtenue sur une page word qui sera imprimée en fin de séance.*



3) Compléter dans le tableau toutes les valeurs des grandeurs qui caractérisent le mouvement de vénus qui s'affichent dans le programme python.

	Demi-grand axe a (u.a)	***Excentricité e (sans unité)***	***Période de révolution T (jours)***
Mercure	*0,387*	*0,206*	*88*
Vénus	***0,723***	***0,068***	***225***
Terre	*1,000*	*0,017*	*365*
Mars	*1,524*	*0,093*	*687*
Jupiter	5,203	0,048	4333
Saturne	9,54	0,056	10759
Uranus	19,18	0,047	30865
Neptune	30,06	0,0090	60188
Pluton	39,44	0,25	90700

4) La forme de la trajectoire visualisée est-elle en accord avec la 1ère loi de Kepler ?

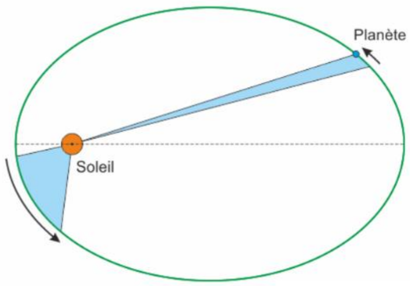
Oui car la trajectoire de Vénus autour du soleil n'est pas un cercle mais une ellipse et le soleil occupe l'un des foyers

5) Commenter les valeurs d'excentricité obtenues pour l'ensemble des planètes du tableau. Quelle approximation peut donc être faite concernant la trajectoire de la plupart des planètes du système solaire ?

Hormis pour Mercure et pluton, les valeurs d'excentricité sont très faibles. On peut donc dire que leurs trajectoires sont quasi-circulaires.

	APPEL	
	Appeler le professeur pour lui présenter les résultats ou en cas de difficulté	

2ème partie : tester la 2ème loi de Kepler



1) Consulter l'animation présentée à l'adresse suivante et cocher « vecteur vitesse »:
   https://www.walter-fendt.de/html5/phfr/keplerlaw2_fr.htm
   Choisir une planète où l'on voit bien une ellipse (type Pluton).

2) Copier le programme suivant et exécuter le.

```
from pylab import *
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
import math as mat

data = pd.read_csv("venus.csv", sep=";",
decimal=".")

x = data['px (au)']
y = data['py (au)']

plt.figure(dpi=180)
plt.xlabel('Coordonnees x')
plt.ylabel('Coordonnees y')
plt.title('Trajectoire de venus')
plt.scatter(x, y, marker = '+', color='purple')
i = 1
while i <200 :
   alpha = arctan2(y[i], x[i]) - arctan2(y[i-1], x[i-1])
   r0 = sqrt(x[i]**2 + y[i]**2)
   r1 = sqrt(x[i-1]**2 + y[i-1]**2)
   A = r0*r1*sin(alpha)/2 # calcul de l'aire
   plt.fill([x[i], x[i-1], 0.25], [y[i], y[i-1], 0], label='A =
' + "%.2e"%A + ' u.a.**2')
   i += 30
plt.legend(loc='center left')
plt.show()
```

3) Que montrent l'animation et le programme ?

Ces outils nous montrent (y compris quantitativement pour vénus) que les aires balayées durant des durées égales sont égales : c'est la 2ème loi de Kepler.

4) Le mouvement de la planète est-il uniforme ? Préciser. Où semblent se situer les extrémums de vitesse.

Non, il n'est pas uniforme. Vitesse maximale au périhélie, et minimale à l'aphélie.

5) Le programme suivant mis à votre disposition ci-dessous doit vous permettre d'étudier plus précisément l'évolution temporelle de la vitesse *V* d'une planète dans le référentiel héliocentrique et de sa distance *d* par rapport au soleil. Comme dans la première partie, ce programme exploitera les données orbitales issues du fichier *.csv* créé précédemment, mais reste à finaliser.
   a. Copier le programme dans l'éditeur Python disponible *(Edupython)*

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
import math as mat

# PARTIE 1 RECUPERATION DONNEES ORBITALES DANS
FICHIER CSV
data = pd.read_csv("venus.csv", sep=";", decimal=".")

X = data['px (au)']
Y = data['py (au)']
Z = data['pz (au)']

n=(len(X))

VX = data['vx (au/d)']
VY = data['vy (au/d)']
VZ = data['vz (au/d)']

# PARTIE 2 CALCUL DE LA VITESSE DE LA PLANETE
V=[]
n=(len(X))
for i in range(0,n-1) :
   V.append(mat.sqrt(VX[i]**2+VY[i]**2+VZ[i]**2))

# PARTIE 3 CALCUL DISTANCE AU SOLEIL
d=[]
for i in range(0,n-1) :
   d.append(mat.sqrt(X[i]**2+Y[i]**2+Z[i]**2))

# PARTIE 4 GRAPHE DE d ET V EN FONCTION DU TEMPS
t=[]
for i in range(0,n-1) :
   t.append(i)

plt.subplot(211)
plt.plot(t,d,"+")
plt.xlabel('Temps (jours)')
plt.ylabel('Distance au soleil (ua)')

plt.plot(t,V,"x")
plt.xlabel('Temps (jours)')
plt.ylabel('vitesse en (ua/jours)')

plt.show()
```

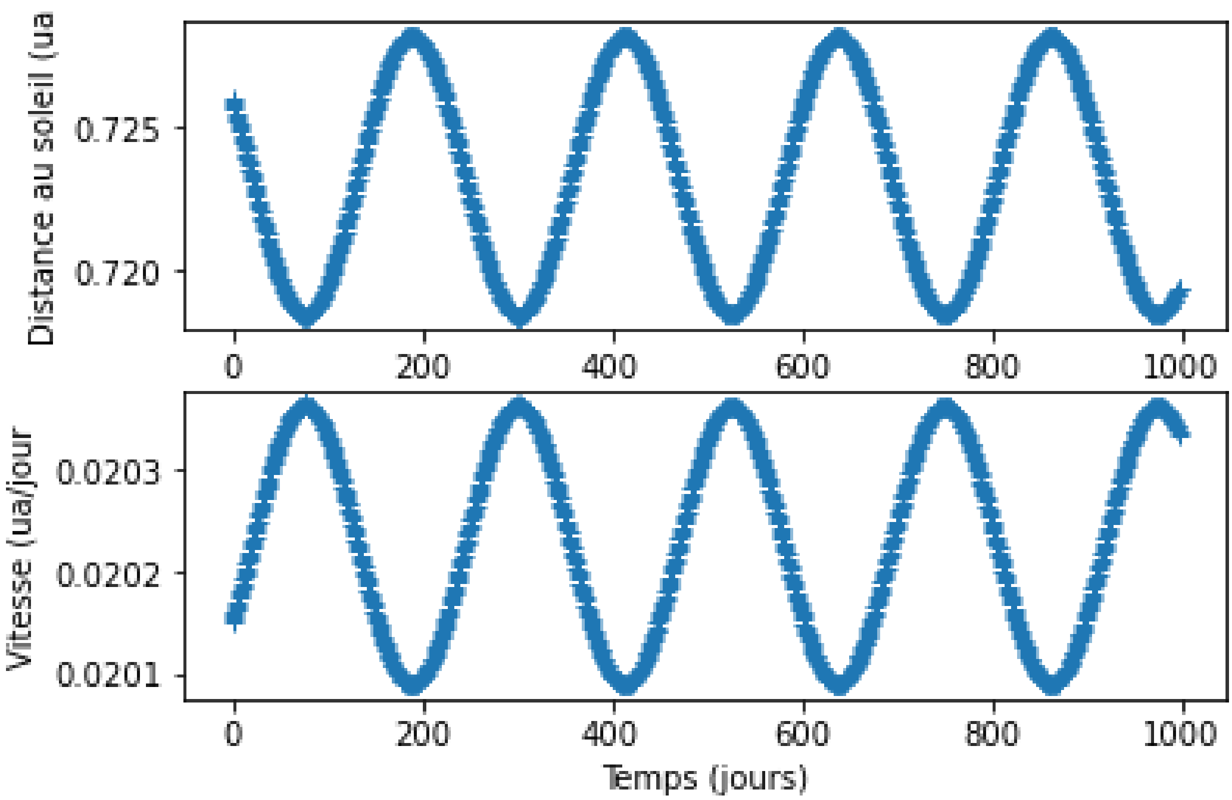
b. Donner l'expression littérale de la vitesse *V* de la planète en fonction de ses coordonnées cartésiennes V_x, V_y, *et* V_z. Compléter alors la ligne rouge du programme avec la formule de calcul de la vitesse *V* de la planète. (Écrire sur le programme)

c. En vous inspirant des lignes **violettes** permettant d'obtenir le tracé du graphe *d = f(t)*, écrire 4 lignes en dessous des lignes violettes permettant d'obtenir le tracé du graphe *V = f(t)*.

d. Exécuter le programme pour observer les 2 graphes en concordance de temps.

e. Quelle est l'allure des graphes obtenus ?

Ces sont des sinusoïdes



f. Quelle observation, conséquence de la 2ème loi de Kepler, et faite à partir de l'animation précédente, est confirmée par ces graphes ?

Les 2 sinusoïdes sont en opposition de phase. Quand la distance au soleil est minimale, la vitesse de la planète est maximale. Quand la distance au soleil est maximale, la vitesse de la planète est minimale. C'est une conséquence de la 2ème loi de Kepler.

	APPEL	
	**Appeler le professeur pour lui présenter les résultats ou en cas de difficulté**	

3ème partie : tester la 3ème loi de Kepler

1) Le programme ci-dessous doit vous permettre valider la 3ème loi de Kepler pour les planètes du système solaire. Pour être opérationnel, ce programme doit être complété avec les valeurs de périodes T et de demi-grand axe a qui figurent dans le tableau établi à la première partie. Comme précédemment, ce programme reste à finaliser.

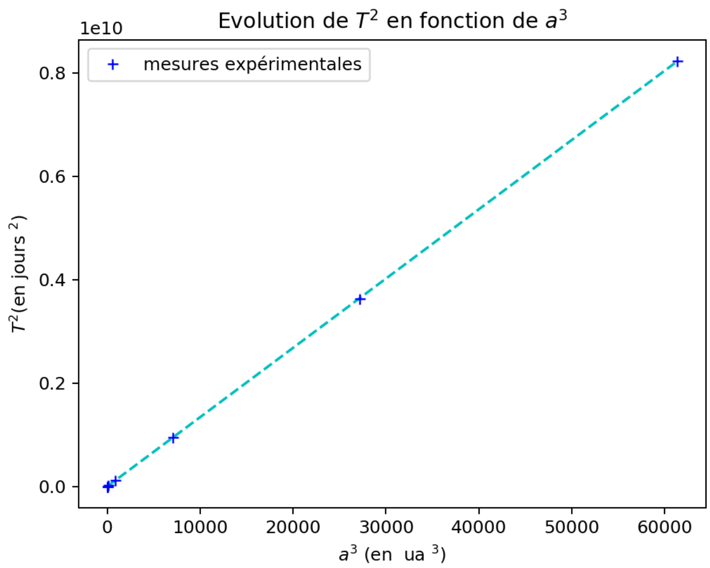
a. Copier le programme 3 dans l'éditeur Python disponible *(Edupython).*

```
from scipy import stats
import matplotlib.pyplot as plt
import statistics as stat

   # CREATION DES LISTES DE DONNEES EXPERIMENTALES

T = [88,225,365,687,4333,10759,30685,60188,90700]
a = [0.387,0.723,1,1.524,5.203,9.54,19.18,30.06,39.44]

   # CALCUL DE T2, a3, ET DU RAPPORT T2/a3
T2 = []
a3 = []
Rapport = []
for i in range(len(a)) :
   T2.append(T[i]**2)
   a3.append(a[i]**3)
   Rapport.append(T2[i]/a3[i])

plt.figure(dpi=180)
plt.title("Evolution de $T^2$ en fonction de $a^3$")
plt.xlabel("$a^3$ (en  ua $^3$)")
plt.ylabel("$T^2$(en jours $^2$)")

   # Modélisation par une fonction linéaire

slope, intercept, r_value, p_value, std_err = stats.linregress(a3, T2)
print("pente: %.3E ordonnée à l'origine: %.3E" % (slope, intercept))
print("R-squared: %f" % r_value**2)
reg=[]
for i in range (len(a3)):
   reg.append(slope*a3[i])
fonction="$T^2$="+ "%.3E" %slope +"* $a^3$ + " + "%.3E" %intercept
a, b, r, _, _ = stats.linregress(a3,T2)
plt.plot(a3, reg, 'c--')
plt.plot(a3, T2,"bo",label='mesures expérimentales', marker='+')
plt.legend()

#TRAITEMENT STATISTIQUES DES RESULTATS

Moyenne = stat.mean(Rapport)

print("\nla liste contenant les valeur du rapport T2/a3 : ")
print(Rapport)
print("\nValeur moyenne = ")
print(Moyenne)
plt.show()
```

b. Saisir les valeurs (en rouge) de périodes T et de demi-grands axes a des différentes planètes qui figurent dans le tableau annexe. *(Respecter le même ordre pour la saisie des 2 listes)*

c. Les variable *T2*, *a3*, et *Rapport* désignent respectivement les grandeurs T^2, a^3, et $\frac{T^2}{a^3}$. Saisir (en rouge) les lignes de commande permettant leur calcul pour chaque éléments T[i] et a[i].

d. Exécuter le programme.




```
pente: 1.340E+05 ordonnée à l'origine: -1.867E+06
R-squared: 0.999995

la liste contenant les valeur du rapport T2/a3 :
[133607.9957622249, 133952.2905520199, 133225.0, 133339
.42350299397, 133295.6265505356, 133321.03941817433,
 133446.27231952926, 133368.3861216157, 134092
.3047224503]

Valeur moyenne =
133516.4821055049
```

2) Que peut-on conclure quant à l'allure de la courbe obtenue sur T^2 et a^3 ?

T^2 est proportionnelle a a^3 car droite passant par l'origine

3) Relever la moyenne du rapport $\frac{T^2}{a^3}$ fourni par le programme ainsi que l'écart-type. La 3ème loi de Kepler vous semble-t-elle validée dans le cas des planètes du système solaire ? Justifier.

Oui le rapport est quasiment constant donc $\frac{T^2}{a^3} = 133516$

Ce résultat confirme donc que le rapport $\frac{T^2}{a^3}$ peut-être considéré comme constant dans le cas de notre étude et que la 3ème loi de Kepler est validée.

4) Grâce à ses travaux sur la gravitation universelle, Newton va réussir à expliciter la constante qui figure dans la 3ème loi de Kepler en démontrant que :

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

Avec G = constante de gravitation Universelle = $6{,}67\text{x}10^{-11}$ SI
Et M = masse de l'astre attracteur en kg.
Estimer la masse M du soleil.

Données : 1 ua = $1.496 \,.\, 10^{11}$ m , G = $6.67 \,.\, 10^{-11}$ SI

$$M = \frac{4\pi^2 a^3}{GT^2} = \frac{4 \times \pi^2 \times (0{,}723 \times 1{,}496.10^{11})^3}{6{,}67.10^{-11} \times (225 \times 24 \times 3600)^2} = 1{,}98.10^{30}\, kg$$

	APPEL	
	Appeler le professeur pour lui présenter les résultats ou en cas de difficulté	

4ème partie : quelques applications (pour les plus rapides)

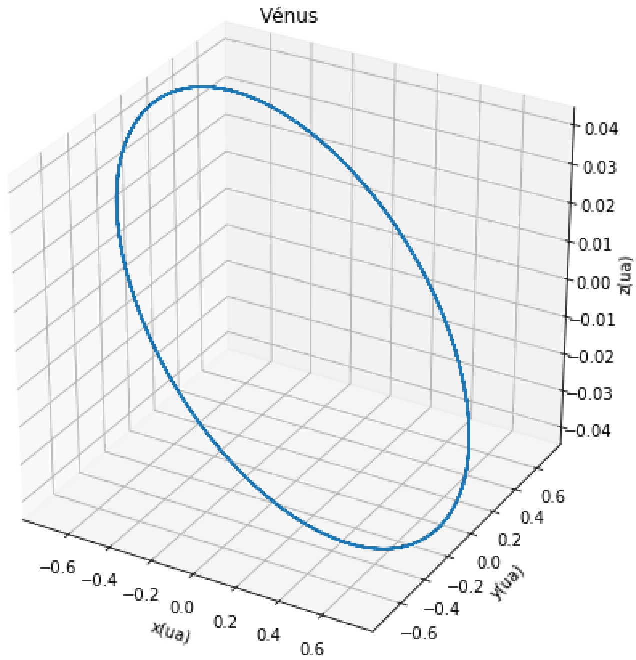
1) Montrer que la comète de Halley, dont la période de révolution est de *76 ans* et la valeur du demi grand axe est de *17,9 ua*, est en orbite autour du soleil.

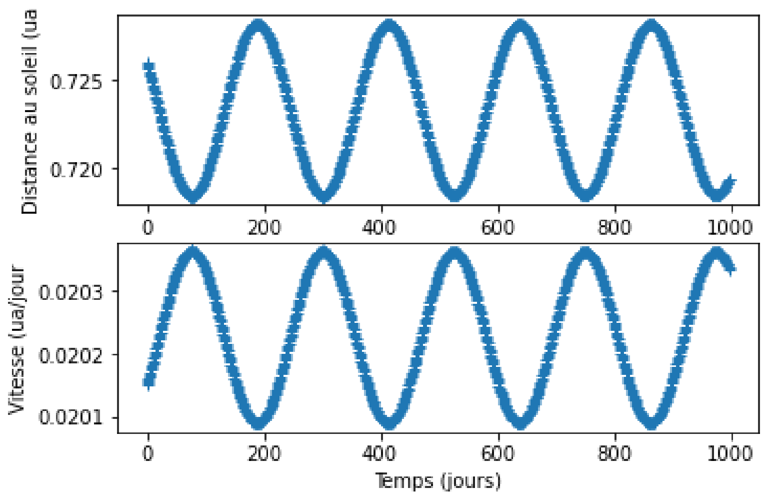
On calcule le rapport $\frac{T^2}{a^3}$ avec T en jours et on montre que c'est le même que pour celui des planètes du système solaire donc la comète est en orbite autour du soleil.

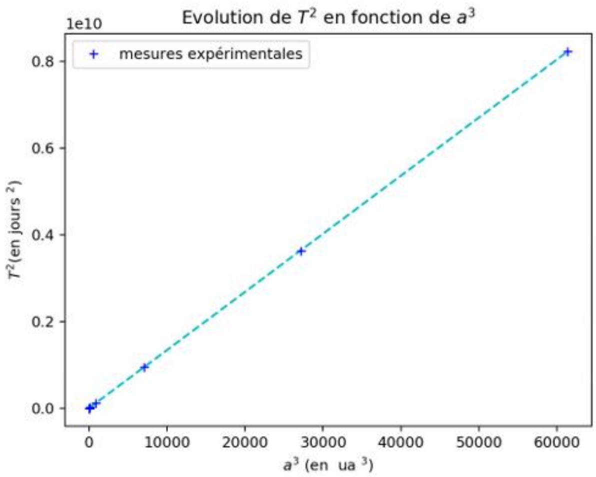
2) La comète Neowise, en orbite autour du soleil, était visible à l'œil nu depuis la Terre. Sa période de révolution a été estimée à 6767 ans par la NASA. Déterminer la valeur a du demi grand axe de sa trajectoire en ua.

$$a = \left(\frac{T^2}{cte}\right)^{1/3} = \left(\frac{(6767 \times 365)^2}{133444}\right)^{1/3} = 357.6ua$$

A IMPRIMER POUR LES ELEVES








```
pente: 1.340E+05 ordonnée à l'origine: -1.867E+06
R-squared: 0.999995

la liste contenant les valeur du rapport T2/a3 :
[133607.9957622249, 133952.2905520199, 133225.0, 133339
.42350299397, 133295.6265505356, 133321.03941817433,
 133446.27231952926, 133368.3861216157, 134092
.3047224503]

Valeur moyenne =
133516.4821055049
```