Partie III Peser la Terre

III.1. Principe

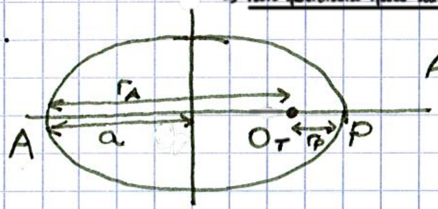
Q22. Express° de l'interac° gravitationnelle entre 2 astres A et B : $\vec{F}_{A\to B} = -G \frac{m_A m_B}{AB^3} \vec{AB}$

↳ fait intervenir les masses m_A et m_B

On parle du poids d'un système de masse m lorsque celui-ci se trouve à proximité de la surface terrestre, on a $\vec{P} = m\vec{g}$ → champ de pesanteur terrestre.

↳ non pertinent pour un astre !

Q23.



$AP = 2a = AO_T + O_TP = r_A + r_P \Rightarrow \boxed{a = \frac{r_A + r_P}{2}}$

Q24 3ᵉ loi de Kepler : $\frac{a^3}{T^2} = \frac{GM_T}{4\pi^2}$ $\boxed{\begin{matrix} a^3 = \alpha T^2 \\ Y = \alpha X \end{matrix}}$ avec $\boxed{Y = a^3}$; $\boxed{X = T^2}$ et $\boxed{M_T = \frac{4\pi^2 \alpha}{G}}$

III 2. Etude de données

Q25 Le code créé une liste contenant les numéros des satellites étudiés en allant les chercher dans la liste DATA.

Q26

```
T_sat = [ ]
for i in range(9):
    T_sat.append(DATA[i][2])
```

Q27

```
def demiGrandAxe(DATA):
    a_sat = [ ]
    for i in range(9):
        a_sat.append(RT + (max(DATA[i][6]) + min(DATA[i][6]))/2)
    return a_sat
```

Q28

```
def XY(LT, La):
    Xlist = [ ]
    Ylist = [ ]
    for i in range(9):
        Xlist.append((LT[i]**2))
        Ylist.append((La[i]**3))
    return(np.array(Xlist), np.array(Ylist))
```

Q29

```
X = XY(T_sat, a_sat)[0]
Y = XY(T_sat, a_sat)[1]

[alpha, beta] = np.polyfit(X, Y, 1)
```

on attend $\beta = 0$ car $\underset{\uparrow \, a^3}{Y} = \alpha \underset{\uparrow \, T^2}{X}$ est linéaire

Q30

A : X
B : Y
C : X
D : Y_mod
E : T∧2 en min∧2
F : a∧3 en km∧3

Q31 nommer les axes et donner les unités
mettre des barres d'erreurs

Q32 les points sont en fait regroupés en 4 lots très voisins

ex 92,9 ; 93 ; 101,9 ; 118,4 ; 104,2 points très proches au carré vu l'échelle

631,1
1214,4
1436,1 ; 1436,2 points très proches vu l'échelle et le carré.

III. 3. Précision du résultat

Q33 N_sim = 1000 (ou 10 000).

Q34
- G : [ ]
- H : [ ]
- I : [ ]
- J : [ ]
- K : [ ]
- L : N_sim
- M : np.array(a_sat)
- N : np.array(T_sat)
- O : a_tir**3
- P : T_tir**2
- Q : Y_tir
- R : X_tir
- S : x_tir
- U : y_tir
- V : p[0]
- W : p[1]
- Z : (p[0]*4*((np.pi)**2)/(g))*((10**9)/(60**2)) ⚠ unités.
- AA : remplissage des listes de X et Y.

Q35
```
M_T = np.mean(list_MT, axis=0)
u_T = np.std(list_MT, axis=0, ddof=1)
beta_sim = np.mean(list_beta)
u_beta = np.std(list_beta, axis=0, ddof=1)
```
non nécessaire

Q36

- résultat de la simulat° : on retient 2CS sur l'incertitude-type : $u\_M_s = 7{,}7 \cdot 10^{20}$ kg
  ce qui impose un nb de CS sur M_T :
  $M\_T_s = 5{,}97581 \cdot 10^{24}$ kg $u\_M_s = 7{,}7 \cdot 10^{20}$ kg
  → intervalle de valeurs acceptables :
  $[M\_T_s - \Delta M_s ; M\_T_s + \Delta M_s]$
  avec $\Delta M_s = \sqrt{3} \cdot u\_M_s = 1{,}3 \cdot 10^{21}$ kg
  $\Rightarrow M\_T_s \in [5{,}9745 ; 5{,}9771] \cdot 10^{24}$ kg

- de $\hat{m}$ avec les données issues de la littérature : $\Delta M_\ell = \sqrt{3} \cdot u\_M_\ell = 1{,}0 \cdot 10^{21}$ kg
  $\Rightarrow M\_T_\ell \in [5{,}9712 ; 5{,}9732]$

→ Bilan : les 2 intervalles n'ont pas d'intersec° commune ⇒ les résultats obtenus à partir de DATA ne sont pas compatibles avec la valeur publiée.

→ Autre réponse possible : calcul de l'écart normalisé (= z-score) :

$$E_N = \frac{|M\_T_\ell - M\_T_s|}{\sqrt{u\_T_\ell^2 + u\_T_s^2}} = \frac{3{,}58 \cdot 10^{21}}{9{,}7 \cdot 10^{20}} = 3{,}7 > 2$$

↳ résultats incompatibles.

Commentaire : les incertitudes sur Data ont peut-être été sous-estimées.