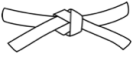





# Gravitation

Travailler avec vos cours et TD ouverts est **chaudement recommandé** : un DM est un entraînement, pas une évaluation. Réfléchir ensemble est une bonne idée, mais le travail de rédaction doit être individuel. En cas de besoin, **n'hésitez pas à me poser des questions**, idéalement à la fin d'un cours ou éventuellement par mail.

Ceinture		Travail à réaliser
	Ceinture blanche	En entier
	Ceinture jaune	En entier
	Ceinture rouge	En entier ou X MP 2019
	Ceinture noire	En entier ou X MP 2019



Flasher ou cliquer pour accéder au corrigé

## Tchouri, Rosetta et Philae

Rosetta est une mission spatiale de l'Agence Spatiale Européenne dont l'objectif était de recueillir des données sur la composition du noyau de la comète 67P/Tchourioumov-Guerassimenko (du nom de ses découvreurs M. Iouri Tchourioumov et Mme Svetlana Guerassimenko), dite plus simplement Tchouri, et sur son comportement à l'approche du Soleil. La sonde spatiale s'est placée en orbite autour de la comète, puis, après une période d'observation de plusieurs mois, a envoyé le 12 novembre 2014 Philae, un petit atterrisseur, se poser sur sa surface pour analyser la composition de son sol et sa structure. L'agence spatiale a mis fin à la mission Rosetta le 30 septembre 2016, en posant l'engin sur le sol de la comète. Elle a largement atteint ses objectifs, et donné lieu à de nombreuses découvertes inédites sur la structure et la composition de la comète.



Données :

- Masse de la comète Tchouri :  $m_T = 1,0 \cdot 10^{13}$  kg ;
- Masse de la sonde Rosetta :  $m_R = 1500$  kg ;
- Masse de l'atterrisseur Philae :  $m_P = 100$  kg ;
- Distance de largage par rapport au centre :  $r_0 = 22,5$  km ;
- Constante gravitationnelle :  $G = 6,7 \cdot 10^{-11}$  m<sup>3</sup> · kg<sup>-1</sup> · s<sup>-2</sup>.

La forme de la comète rend son champ gravitationnel impossible à exprimer simplement. Pour simplifier, on la modélise par une boule homogène de masse  $m_T$  et de rayon  $r_T = 2 \cdot 10^3$  m, ce qui permet d'utiliser les expressions usuelles pour la force et l'énergie potentielle gravitationnelles. La distance entre un point  $M$  et le centre  $O$  de la comète est notée  $r = OM$ .

## A - Rosetta tourne autour de Tchouri

La sonde Rosetta a commencé par se rapprocher par paliers de la comète<sup>1</sup>. Le 10 septembre 2014, elle se situait sur une première orbite approximativement circulaire de rayon  $r_1 = 30$  km.

1 - Établir l'expression de la vitesse  $v_1$  de la sonde en orbite circulaire de rayon  $r_1$  autour de la comète en fonction de  $G$ ,  $m_T$  et  $r_1$ . En déduire sa période  $T_1$ . Les calculer numériquement.

Pour se rapprocher davantage de la comète, la sonde est placée à partir du 8 octobre 2014 sur une orbite elliptique<sup>2</sup> avec un apocentre  $A$  situé à la distance  $r_A = r_{\max} = 20$  km du centre  $O$  de la comète et un péricentre  $P$  tel que  $r_P = r_{\min} = 10$  km. Le 15 octobre, la propulsion est utilisée pour placer la sonde sur une orbite circulaire de rayon  $r_P$ .

2 - Représenter sur un schéma l'orbite elliptique, en faisant apparaître le centre  $O$  de la comète, les points  $A$  et  $P$ , et les distances  $r_A$  et  $r_P$ .

3 - Établir l'expression de l'énergie mécanique de la sonde sur l'orbite elliptique en fonction de  $r_A$  et  $r_P$ .

4 - Sur cette orbite, en déduire la vitesse  $v_P$  de Rosetta en  $P$ .

5 - Pour placer la sonde en orbite circulaire de rayon  $r_P$ , la propulsion est utilisée lorsque Rosetta est au péricentre. Déterminer littéralement et numériquement la variation de vitesse nécessaire.

Après cette phase d'exploration au plus près de la comète, la sonde Rosetta est remontée sur une orbite quasi-circulaire de rayon de l'ordre de 20 km pour pouvoir larguer la sonde Philae. Contrairement à ce que suggère l'intuition, remonter en altitude permettait un largage plus sécurisé : en raison de la forme irrégulière de la comète, la trajectoire en orbite très basse était chaotique, rendant difficile le contrôle des conditions de largage.

## B - Philae se fait larguer

On s'intéresse dans cette partie au mouvement de l'atterrisseur Philae lui ayant permis d'atterrir sur la comète après son largage le 12 novembre 2014. On étudie la chute libre de l'atterrisseur Philae dans un référentiel dont l'origine est le centre  $O$  de la comète et qui tourne avec Rosetta<sup>3</sup>, qui peut être considéré comme galiléen. Le vecteur  $\vec{e}_r$  pointe constamment de l'atterrisseur vers le centre de la comète, si bien que l'accélération de Philae peut s'écrire  $\vec{a} = \ddot{r}\vec{e}_r$ .

6 - Établir l'équation du mouvement de l'atterrisseur Philae lors de sa chute.

Cette équation n'admet pas de solution analytique simple, mais peut être résolue numériquement. On se propose de le faire à l'aide de la fonction `odeint` du module `scipy.integrate`, dont l'utilisation est rappelée en fin d'énoncé. On pose  $v = \dot{r}$  et  $y(t) = [r(t), v(t)]$ .

7 - Écrire le système différentiel du premier ordre vérifié par les deux composantes du vecteur  $y$ .

8 - Compléter les blocs de code associés aux lignes 23, 26 et 30 du code ci-dessous. Expliquer le rôle du test réalisé ligne 20.

```

1  import numpy as np
2  import scipy.integrate as sci
3  import matplotlib.pyplot as plt

4
5  ### Constantes physiques
6  G = 6.7e-11 # m3.kg-1.s-2
7  m_T = 1e13 # kg

8
9  ### Conditions initiales
10 r0 = 22.5e3 # m
11 v0 = -.15 # m.s-1
12 y0 = [r0, v0]
```

1. Notez que c'est un exploit en soi ! La précision requise pour l'accomplir est de quelques kilomètres pour des objets évoluant sur des orbites gravitationnelles à des vitesses de l'ordre du kilomètre par seconde, à plusieurs centaines de millions de kilomètres de la Terre.

2. À cause de la forme irrégulière de la comète, les trajectoires réelles sont beaucoup plus compliquées et nécessitent des corrections très régulières avec intervention du moteur de la sonde.

3. Il s'agit plus ou moins d'un équivalent du référentiel terrestre, avec quelques complications dues à la vitesse de rotation propre de la comète qui n'est pas négligeable.

```

14  ### Intervalle de temps pour la simulation
15  t = np.linspace(0, 160e3, 1000) # s

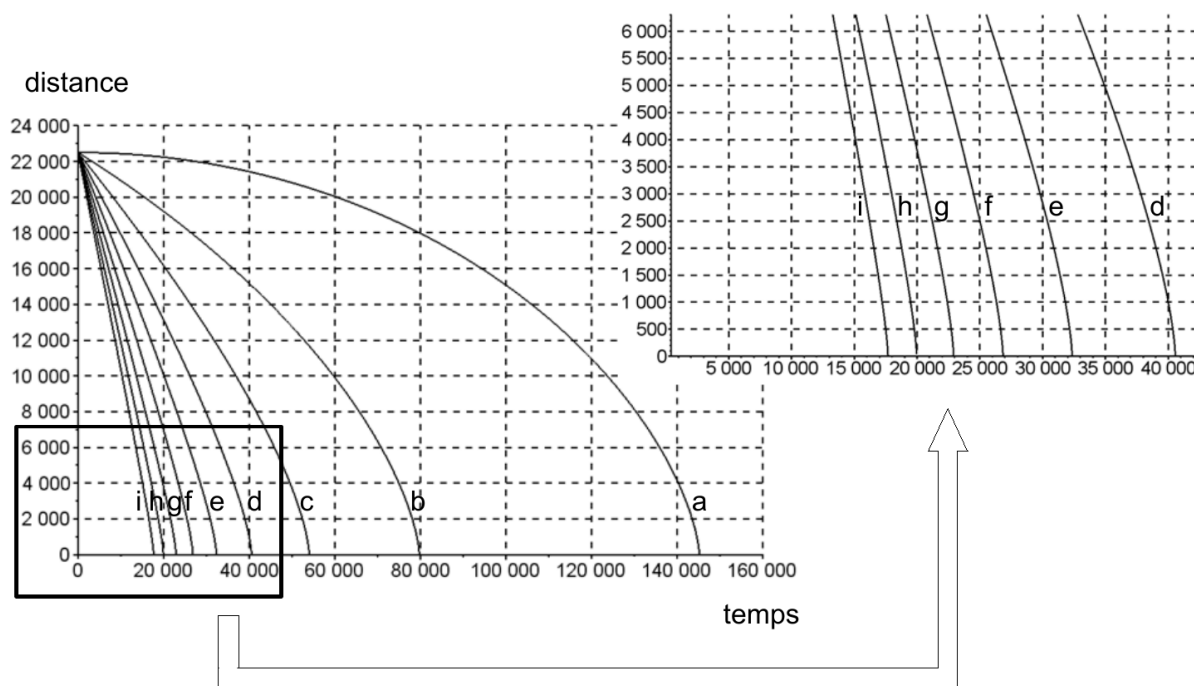
17  ### Fonction d'évolution
18  def f(y, t):
19      r, v = y
20      if r <= 0:
21          return [0, 0]
22      else:
23          # >>> bloc à compléter 1 <<<

25  ### Calcul de la solution
26  # >>> bloc à compléter 2 <<<
27  r = sol[:, 0]

29  ### Tracé de r(t)
30  # >>> bloc à compléter 3 <<<

```

L'évolution temporelle de la distance  $r$  calculée par le code ci-dessus est représentée sur la figure 1 pour différentes conditions initiales.



**Figure 1 – Évolution temporelle de l'altitude pour différentes vitesses initiales.** Temps en s, distance en m.  
(a)  $v_0 = 0 \text{ m} \cdot \text{s}^{-1}$ ; (b)  $v_0 = -0,15 \text{ m} \cdot \text{s}^{-1}$ ; (c)  $v_0 = -0,30 \text{ m} \cdot \text{s}^{-1}$ ; (d)  $v_0 = -0,45 \text{ m} \cdot \text{s}^{-1}$ ; (e)  $v_0 = -0,60 \text{ m} \cdot \text{s}^{-1}$ ; (f)  $v_0 = -0,75 \text{ m} \cdot \text{s}^{-1}$ ; (g)  $v_0 = -0,90 \text{ m} \cdot \text{s}^{-1}$ ; (h)  $v_0 = -1,05 \text{ m} \cdot \text{s}^{-1}$ ; (i)  $v_0 = -1,20 \text{ m} \cdot \text{s}^{-1}$ .

- 9 - Déterminer la durée  $\tau_0$  de la chute de Philae s'il est abandonné par Rosetta avec une vitesse verticale nulle.
- 10 - La durée réelle de la chute est d'environ sept heures. En déduire la vitesse initiale communiquée à l'atterrisseur.
- 11 - Le code permet également de représenter figure 2 l'évolution de la vitesse de Philae en fonction de son altitude. Déterminer, par lecture graphique, la vitesse verticale atteinte par Philae au moment du contact avec la comète.
- 12 - On cherche à interpréter cette valeur. Comment évolue l'énergie mécanique de l'atterrisseur au cours de sa chute? En déduire la vitesse au moment du contact avec la comète et la calculer numériquement. Est-elle cohérente avec la résolution numérique?

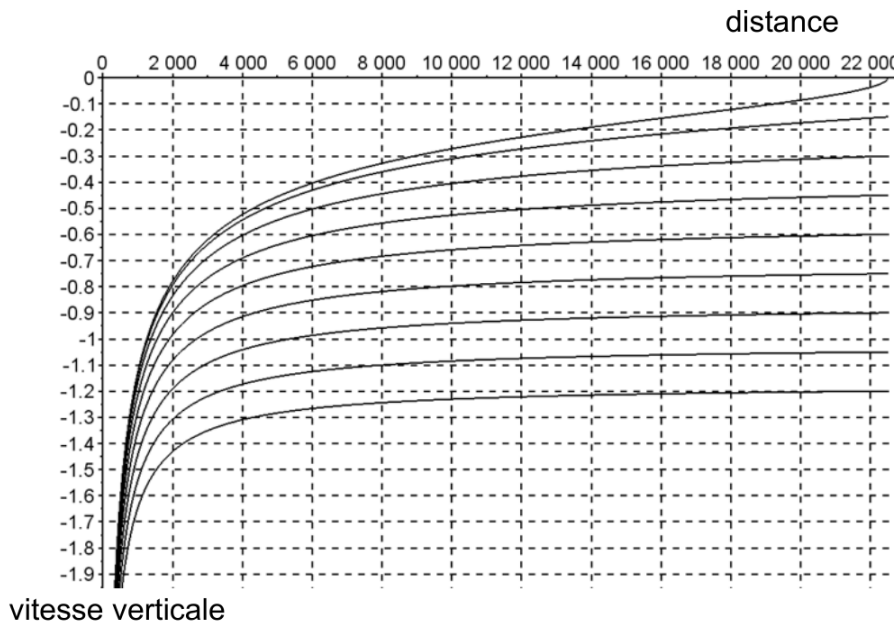


Figure 2 – Vitesse en fonction de la distance au centre de la comète. Distance en m, vitesse en  $\text{m} \cdot \text{s}^{-1}$ .

### Annexe : Utilisation de `scipy.integrate.odeint`

La fonction `odeint` de la bibliothèque de calcul scientifique SciPy (disponible dans le module `scipy.integrate`) est couramment utilisée pour la résolution numérique d'un système d'équations différentielles ordinaires qui s'écrit sous la forme

$$\frac{dy}{dt} = f(y, t)$$

dont l'inconnue  $y$  peut être un vecteur (cas du sujet). La syntaxe de base pour la résolution numérique de l'équation est la suivante :

```
1 | sol = scipy.integrate.odeint(func, y0, t)
```

Les principaux paramètres sont

- `func` : fonction Python qui effectue le calcul numérique de la fonction  $f$  du système différentiel. Elle prend comme arguments, dans cet ordre,
  - `y` : liste ou tableau NumPy de flottants correspondant à  $y$  ;
  - `tc` : flottant correspondant à une valeur  $t_c$  de  $t$ .
- `y0` : liste ou tableau NumPy contenant la condition initiale  $y(0)$  relative à l'inconnue  $y$  du système différentiel ;
- `t` : tableau NumPy définissant les valeurs de la variable  $t$  pour lesquelles on souhaite obtenir les valeurs numériques pour l'inconnue  $y$  du système différentiel.

La fonction `odeint` renvoie un tableau de flottants de taille  $N_t \times N_y$ , où  $N_t$  est le nombre d'éléments du tableau `t` et  $N_y$  le nombre d'éléments du tableau `y0`. Il contient les valeurs numériques des  $y(t)$  solutions pour les valeurs de  $t$  contenues dans le tableau `t`.