TP 28 : Simulation de trajectoires dans un champ de force conservatif
Méthodes numériques

Les points du programme :

- Capacité numérique : *à l'aide d'un langage de programmation, obtenir des trajectoires d'un point matériel soumis à un champ de force centrale conservatif.*

Domaines numériques	Capacités exigibles
Outils graphiques	
Représentation graphique d'une fonction.	Utiliser les fonctions de base de la bibliothèque **matplotlib** pour tracer la courbe représentative d'une fonction.
Équations différentielles	
Equations différentielles d'ordre supérieur ou égal à 2	Transformer une équation différentielle d'ordre n en un système différentiel de n équations d'ordre 1. Utiliser la fonction **odeint** de la bibliothèque **scipy.integrate** (sa spécification étant fournie).

On a déjà vu comment obtenir numériquement une solution approchée d'une équation différentielle par la méthode d'Euler.
En fait, le module Python appelé scipy.integrate contient une fonction, appelée odeint (pour « ***o****rdinary* ***d****ifferential* ***e****quation* ***int****egration* »), qui permet de résoudre directement l'équation sans avoir à programmer soi-même la méthode de résolution.
Les avantages sont que :

- on gagne du temps ;
- la résolution est en général meilleure qu'avec la méthode d'Euler « basique » car odeint utilise des schémas de résolution un peu plus sophistiqués que le schéma d'Euler explicite (même si les principes de base sont les mêmes).

Utilisation de la fonction odeint

Il est possible de résoudre efficacement un système d'équations différentielles du premier ordre du type

$$\frac{dY}{dt} = F(Y,t) \quad \xrightarrow{ordre\ 2} \quad \frac{d}{dt}\begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix} = F\left(\begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix}, t\right)$$

à l'aide la fonction `odeint` de la librairie `scipy.integrate` dont voici la syntaxe :

```
from scipy.integrate import odeint
Y_sol = odeint(F,Y0,liste_t)
```

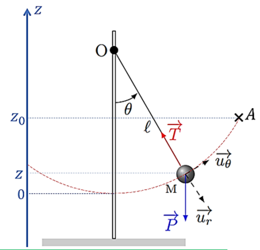
Paramètres de odeint :

- `F` est une fonction du type `F(Y,t)`, qui renvoie la valeur de la dérivée de y à l'instant `t`.
  Pour résoudre une équation d'ordre 1, y est un scalaire.
  Mais si l'équation est d'ordre 2, alors y est de type vecteur : $Y = \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix}$.
  Attention, même si `F` ne dépend pas de `t`, il faut tout de même que `t` apparaisse comme second argument : `F(Y,t)`.
- `Y0` : valeur initiale de Y. C'est un couple qui a autant de composantes que Y.
- `liste_t` est une liste qui contient les instants auxquels la solution sera calculée.

Après exécution de `Y_sol = odeint(F,Y0,liste_t)` :

- `Y_sol` est une matrice dont la première colonne contient les valeurs de $Y_0(t)$ et la seconde colonne contient les valeurs de $Y_1(t)$.

1. Rappel : Pendule simple



On souhaite simuler le comportement d'un pendule simple. On rappelle l'équation différentielle qui régit son comportement (équation du mouvement) :

$$\ddot{\theta} + {\omega_0}^2.\sin(\theta) = 0$$

avec $\omega_0 = \sqrt{g/\ell}$.
Cette équation est une équation d'**ordre 2**.

➔ Il faut d'abord de transformer cette équation en un système de 2 équations du 1er ordre ! On utilise pour cela la grandeur intermédiaire $\dot{\theta}$. On peut alors écrire :

$$\begin{cases} \dfrac{d\theta}{dt} = \dot{\theta} \\ \dfrac{d\dot{\theta}}{dt} = -{\omega_0}^2.\sin(\theta) \end{cases}$$

Pour utiliser la fonction odeint, les composantes du vecteur Y sont ici θ et $\dot{\theta}$: $Y = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$

- ☐ Récupérer dans le dossier Mécanique/M6/TP28 du cahier de prépa le fichier *TP28_Pendule_odeint.py*. L'ouvrir avec Spyder.
- ☐ Compléter dans le script la fonction F afin qu'elle décrive le comportement du pendule.
- ☐ Exécuter le script et vérifier que le résultat est celui attendu.

En présence de frottements proportionnels à la vitesse, l'équation du mouvement devient :

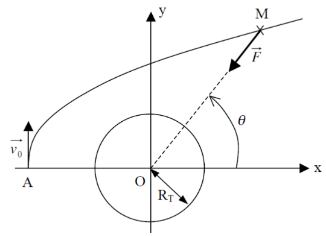
$$\ddot{\theta} + \frac{k}{m}.\ell.\dot{\theta} + {\omega_0}^2.\sin(\theta) = 0$$

Où k est un facteur caractéristique des frottements. On prendra $k = 0.1$ kg.m^{-1}.s^{-1}.

- ☐ Modifier le script afin de prendre en compte les frottements.
- ☐ Exécuter le script et vérifier que le résultat est cohérent avec celui attendu pour un système en présence de frottements.

2. Application à l'étude des trajectoires des satellites terrestres

2.1. Etude théorique



On considère un satellite M de masse m initialement au point A de coordonnées $x_A = -8400$ km (i.e. à 2000 km de la surface terrestre) et $y_A = 0$. On lui communique alors une vitesse initiale v_0 dirigée selon u_y. On se demande, en fonction de la valeur de v_0, quelle sera la nature de sa trajectoire. Pour chaque question, vous donnerez une expression littérale et une valeur numérique.

Données :

- *Masse de la Terre :* $M_T = 5{,}97.10^{24}\ kg$
- *Rayon de la Terre :* $R_T = 6370\ km$
- *Constante de gravitation universelle :* $G = 6{,}67.10^{-11}$ u.S.I.

Q1. Pour quelle valeur $v_{0,c}$ de v_0 le satellite entrera-t-il sur une orbite circulaire ?

Q2. Pour quelle valeur minimale $v_{0,lib}$ de v_0 le satellite va-t-il pouvoir s'échapper de l'attraction terrestre ? (vitesse de libération)

Q3. Quelle est la nature des trajectoires pour $v_{0,c} < v_0 < v_{0,lib}$? Pour $v_0 > v_{0,lib}$?

2.2. Résolution numérique avec Python

En coordonnées polaires dans le plan du mouvement, le PFD appliqué au satellite assimilé à un point matériel M donne :

$$m.\frac{d\overrightarrow{OM}}{dt} = -G.\frac{m.M_T}{r^2}.\overrightarrow{u_r} = K.\frac{m}{r^2}.\overrightarrow{u_r} \quad \text{avec } K = -G.M_T$$

On utilise en plus : $\overrightarrow{u_r} = \frac{\overrightarrow{OM}}{\|\overrightarrow{OM}\|} = \frac{x.\overrightarrow{u_x}+y.\overrightarrow{u_y}}{r}$ avec $\|\overrightarrow{OM}\| = r = \sqrt{x^2+y^2}$.

On obtient ainsi en projetant sur $\overrightarrow{u_x}$ et $\overrightarrow{u_y}$ les équations du mouvement suivantes :

$$\ddot{x} = -G.\frac{M_T}{r^3}.x = K.\frac{x}{\left(\sqrt{x^2+y^2}\right)^3} \qquad \text{et} \qquad \ddot{y} = -G.\frac{M_T}{r^3}.y = K.\frac{y}{\left(\sqrt{x^2+y^2}\right)^3}$$

Q4. On pose alors $Y = \begin{pmatrix} Y_0 = x \\ Y_1 = y \\ Y_2 = \dot{x} \\ Y_3 = \dot{y} \end{pmatrix}$. Montrer en dérivant Y que l'on obtient :

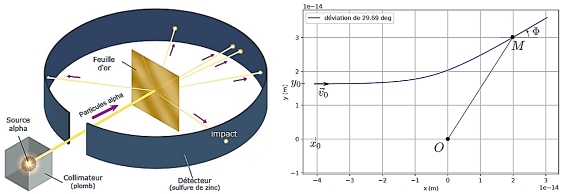
$$\frac{dY}{dt} = F(Y) \qquad \text{avec } F(Y) = \left(Y_2, Y_3, \frac{K.Y_0}{\left(\sqrt{Y_0^2+Y_1^2}\right)^3}, \frac{K.Y_1}{\left(\sqrt{Y_0^2+Y_1^2}\right)^3}\right)$$

- ☐ Récupérer dans le dossier Mécanique/M6/TP28 du cahier de prépa le fichier *TP28_satellite.py*. L'ouvrir avec Spyder.
- ☐ Compléter le code de la fonction $F(Y,t)$ pour qu'il soit en accord avec la relation que l'on vient d'obtenir.
- ☐ Vérifier vos réponses aux questions Q1 à Q3 à l'aide du programme.
- ☐ Utiliser le programme pour simuler la situation de l'exercice 4 du TD « Astéroïde géocroiseur ».

3. L'expérience de Rutherford

On s'intéresse maintenant à une expérience de type Rutherford : un noyau α (composé de deux protons et deux neutrons) est envoyé sur une fine feuille d'or. Il s'agit donc de modéliser la collision entre :

- le noyau α (point M, masse m = 4×1,67.10^{-27} kg, charge q_1 = 2×1,6.10^{-19} C) ;
- le noyau de l'atome d'or (à l'origine O, masse si importante qu'il ne bouge pas, charge $q_2 = Z$ × 1,6.10^{-19} C avec Z = 79).



- ☐ Récupérer dans le dossier Mécanique/M6/TP28 du cahier de prépa le fichier *TP28_Rutherford.py*. L'ouvrir avec Spyder.
- ☐ Compléter ce fichier en adaptant ce qui a été fait pour les satellites (seule l'expression de K diffère).

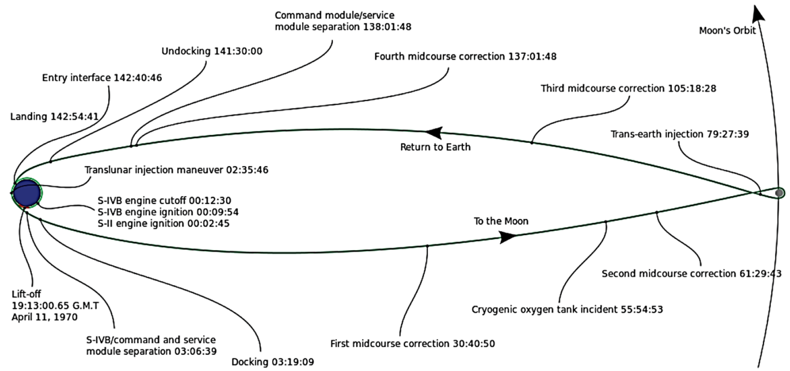
Pour matérialiser la position du noyau, il y a déjà une ligne de code qui trace un cercle de rayon R_0.

- ☐ Explorer l'influence de la valeur initiale $y_0 = y(t=0)$ (aussi appelée paramètre d'impact) sur l'allure de la trajectoire. On pourra tester $y_0 = 2 * R_0$, ou $1 * R_0$, etc. En particulier, donner la valeur de y_0 (en termes de nombre de fois R_0) à partir de laquelle le noyau α est dévié vers l'arrière.

4. Mission Apollo 13 : Système Terre-Lune

Apollo 13 (11 avril 1970 - 17 avril 1970) est une mission lunaire habitée du programme Apollo interrompue à la suite de l'explosion d'un réservoir d'oxygène au cours du trajet vers la Lune. Ne pouvant faire demi-tour, le vaisseau et son équipage furent obligés de poursuivre leur voyage jusqu'à la Lune et d'en faire le tour pour utiliser son attraction gravitationnelle afin de revenir vers la Terre.
L'expression « *Houston, we've had a problem* » qui fut prononcée par l'astronaute Jack Swigert avec un calme impressionnant (en sachant qu'une panne à 300 000 mètres de la Terre a de fortes chances d'être fatale), est vite entrée dans la culture américaine.

d'après Wikipédia



Trajectoire de la mission Apollo 13, détaillant les divers évènements qui se sont produits.

On va maintenant rajouter la Lune dans notre étude et chercher des « trajectoires de retour libres » similaires à celle qui a permis aux astronautes de la mission Apollo 13 de rentrer sur Terre sains et saufs alors qu'ils n'avaient essentiellement plus de moteurs.

<u>Données supplémentaires</u> :

- *Rayon de la Lune : R_L = 1737 km*
- *Masse de la Lune : M_L = 7,35.10^{22} kg*
- *Distance Terre - Lune : D = 384 400 km*

Q5. Ecrire les nouvelles équations différentielles satisfaites par l'abscisse $x(t)$ et l'ordonnée $y(t)$ du vaisseau, en tenant compte maintenant de l'attraction terrestre et de l'attraction lunaire (dans un premier temps, on suppose que la Lune et la Terre sont toutes les deux fixes).

Q6. Modifier votre programme Python précédent pour résoudre ces équations différentielles. Rajouter également le tracé du cercle représentant la Lune.

On suppose que le vaisseau spatial se trouve initialement au point A de coordonnées (x_A = -16 400 km, y_A = 0), c'est à dire qu'il est sur l'axe des abscisses, à gauche de la terre, à une altitude de 10 000 km par rapport au sol (ce qui correspond à une orbite terrestre « moyenne », plus basse que l'orbite géostationnaire).

Q7. Quelle vitesse initiale v_0 (dirigée selon $\overrightarrow{u_y}$) faut-il lui communiquer afin qu'il entre sur une « trajectoire de retour libre » (voir schéma) qui lui permet de faire le tour de la Lune et de rentrer sur Terre en utilisant la force gravitationnelle de la Lune (idéalement, il faudrait qu'à l'arrivée le vaisseau « s'écrase » sur la surface terrestre).

Q8. Combien de temps le vaisseau spatial met-il pour parcourir sa trajectoire ?