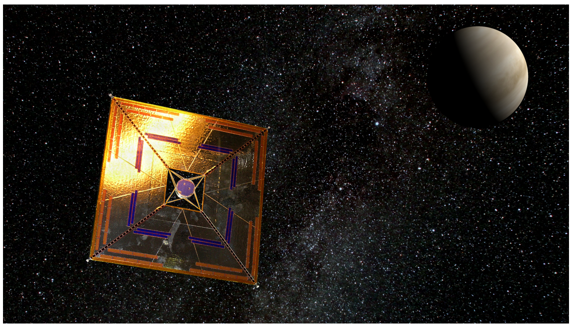
Résolution numérique d'équations différentielles

Techniques et méthodes à acquérir

- Résoudre numériquement une équation différentielle d'ordre 2.
- Utiliser la fonction `odeint` de la bibliothèque `scipy.integrate`.
- Obtenir numériquement des trajectoires dans un champ de force centrale conservatif.

I - Voile solaire



Les voiles solaires visent à propulser des engins spatiaux sans carburant en exploitant la pression de radiation, une force exercée par la lumière lorsqu'elle se réfléchit sur un métal. Compte tenu de la faible propulsion générée, le procédé ne permet pas de quitter la surface d'une planète, même dénuée d'atmosphère. Il est en revanche utilisable sur un appareil ayant déjà atteint la vitesse de satellisation minimale, voire la vitesse de libération. La mission IKAROS (vue d'artiste ci-contre), lancée en 2010, a mis en évidence une accélération due au rayonnement solaire de l'ordre de $10^{-7}\ \mathrm{m \cdot s^{-2}}$, très faible … mais constante.

Ce TP commence par étudier numériquement l'orbite elliptique d'une sonde spatiale dans le champ gravitationnel terrestre, puis introduit un modèle simplifié de voile solaire. L'objectif est de déterminer dans quelle mesure la force exercée par la voile peut modifier la trajectoire de la sonde et permettre une éventuelle évasion du champ gravitationnel.

On travaille en coordonnées cartésiennes. Pour faciliter les calculs numériques, on souhaite adimensionner le problème : le rayon terrestre R_T est pris comme unité de longueur, et la vitesse initiale v_0 de la sonde comme unité de vitesse.

1 - Identifier l'unité de temps τ, puis montrer que l'équation du mouvement adimensionnée portant sur $\vec{r} = \overrightarrow{OM}/R_\mathrm{T}$ s'écrit

$$\ddot{\vec{r}} = -\mu \frac{\vec{r}}{r^3} \qquad \text{avec} \qquad \mu = \frac{G M_\mathrm{T}}{R_\mathrm{T} {v_0}^2}\,,$$

où l'on note d'un point la dérivée par rapport au temps adimensionné.

2 - Écrire le système différentiel du premier ordre vérifié par les coordonnées cartésiennes adimensionnées de la sonde : x, y, v_x, v_y.

Nous allons résoudre ce système différentiel avec la fonction `odeint` du module `scipy.integrate`, dont on rappelle la spécification ci-dessous. Le vecteur Y compte ici quatre composantes : $Y(t) = [x(t), y(t), v_x(t), v_y(t)]$. On prendra comme intervalle de temps `t = np.linspace(0, 20, 1000)`. Le paramètre caractéristique `mu` sera défini comme une variable globale [1].

1. Le passer en argument aux fonctions serait nécessaire pour pouvoir superposer plusieurs trajectoires sur une même figure, et serait globalement plus élégant, mais cela complexifie sensiblement la syntaxe pour pas grand chose dans le contexte très limité de ce TP.

`odeint(f,Y0,t)`

Paramètres d'entrée :
`f` : fonction de Y et du temps t qui renvoie la dérivée $\dot{Y}$ du vecteur d'état Y à l'instant t sous forme d'une liste de même longueur que Y.
`Y0` : vecteur condition initiale $Y(t=0)$.
`t` : tableau des instants pour lesquels doit être calculée la valeur de $Y(t)$.

Valeur de retour :
Tableau contenant autant de colonnes que $Y(t)$ et dont les lignes correspondent aux valeurs aux différents instants du tableau `t`.

3 - Coder la fonction d'évolution `f_sans_voile`, prenant comme arguments le vecteur d'état `Y`, un temps `t` et renvoyant la dérivée de `Y`.

Au moment où elle déploie sa voile, la sonde spatiale doit se trouver sur une orbite basse moyenne que l'on commence par étudier. On suppose l'une des apsides de l'orbite (c'est-à-dire son apogée ou son périgée) à une distance adimensionnée égale à 2. Ainsi, on prend la position initiale au point de coordonnées $(2, 0)$ et la vitesse initiale égale à $(0, 1)$.

4 - Écrire le code permettant de résoudre l'équation du mouvement en faisant appel à `odeint`, c'est-à-dire de récupérer les deux coordonnées adimensionnées $x(t)$ et $y(t)$ de la sonde.

5 - Représenter les trajectoires pour plusieurs valeurs de μ comprises p.ex. entre 0,25 et 4. Pour faire joli, ajouter à la figure un cercle représentant la Terre (de rayon 1 dans les unités naturelles) :

```
theta = np.linspace(0, 2*np.pi, 200)
x_Terre = np.cos(theta)
y_Terre = np.sin(theta)
plt.plot(x_Terre, y_Terre,'k')
```

Si le cercle est aplati car les axes ne sont pas identiques, rajouter la ligne de code `plt.axis('equal')`.

6 - Interpréter la nature des orbites précédentes. La question n'est pas de *constater* que telle orbite semble elliptique et telle autre ouverte, mais de l'*expliquer*.

On prend désormais en compte l'influence de la voile solaire. Les rayons du Soleil arrivant quasi-parallèles, on la modélise en première approche par une force constante, d'intensité décrite par un paramètre adimensionné α :

$$\vec{f} = \alpha \frac{m v_0^2}{R_\mathrm{T}} \vec{e}_x \,.$$

7 - Réécrire le système différentiel et la fonction `f_avec_voile` permettant d'étudier les trajectoires en fonction de μ et α, toujours définies comme des variables globales.

8 - Représenter quelques trajectoires pour $\mu = 1{,}5$ et différentes valeurs de $\alpha \gtrsim 0{,}1$. Augmenter la durée simulée si nécessaire. Qu'en pensez-vous ? Représenter l'évolution de l'énergie mécanique au cours du temps, et proposer une interprétation.

9 - Proposer au moins deux raisons pour lesquelles ce modèle de force est peu réaliste. Proposer des améliorations (simples ...) et les implémenter.