TP11 – Pendule non-linéaire

En dépit de son nom, le pendule simple ne présente pas de solution analytique simple en dehors de l'approximation des petits angles. En effet, l'équation différentielle vérifiée par l'angle θ est **non-linéaire**, ce qui complique sa résolution.

On se propose ici d'étudier le comportement du pendule en dehors de l'approximation des petits angles à l'aide d'une simulation numérique.

Objectifs

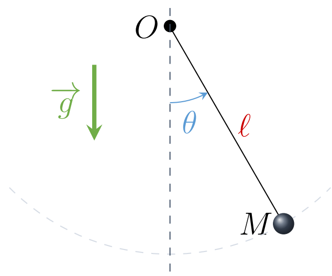
→ Transformer une équation différentielle d'ordre n en un système différentiel de n équations d'ordre 1.

→ Utiliser la fonction `odeint` de la bibliothèque `scipy.integrate` (sa spécification étant fournie).

→ **À l'aide d'un langage de programmation, résoudre numériquement une équation différentielle du deuxième ordre non-linéaire et faire apparaître l'effet des termes non-linéaires.**

Position du problème

Dans toute la suite, la notation θ (dans le texte) ou `theta` (dans le programme) fait référence à l'angle utilisé pour repérer le point M dans le système de coordonnées polaires et $\omega = \dot{\theta}$ ou `omega` à sa vitesse angulaire.



On s'intéresse au mouvement d'une masse m assimilée à son centre de masse M, suspendue à un fil inextensible et de masse négligeable. L'étude est réalisée dans le référentiel terrestre considéré galiléen. On rappelle que l'angle $\theta(t)$ vérifie l'équation différentielle

$$\ddot{\theta} + \frac{g}{\ell}\sin\theta = 0, \qquad (1)$$

et que les éléments cinématiques associés au point M, dans la base polaire s'écrivent :

$$\overrightarrow{OM} = \ell\overrightarrow{e_r} \qquad \vec{v} = \ell\omega\overrightarrow{e_\theta} \qquad \vec{a} = \ell\dot{\omega}\overrightarrow{e_\theta} - \ell\omega^2\overrightarrow{e_r}$$

Le sujet est accompagné de deux programmes, accessibles sur cahier-de-prépa :

- `tp11_pendule.py` : le programme qui sert de point de départ, à modifier ;
- `tp11_fonctions.py` : il contient quelques fonctions utiles pour certaines questions.

Échauffement

On s'intéresse tout d'abord à l'équation différentielle associée au pendule simple pour des oscillations de faible amplitude, où l'on retrouve l'équation différentielle d'un oscillateur harmonique :

$$\ddot{\theta} + \frac{g}{\ell}\theta = 0.$$

Le point M est lâché sans vitesse initiale depuis un angle θ_0.

Le programme `tp11_pendule.py` intègre numériquement cette équation différentielle à l'aide de la fonction `odeint` et représente graphiquement l'évolution temporelle de $\theta(t)$.

1. Modifier le programme pour donner un titre pertinent au graphique et à ses axes.
2. Déterminer graphiquement la période T_0 des oscillations.
3. Indiquer la signification physique de chacune des valeurs de la liste `V0` (ligne 41).
4. Modifier la valeur initiale θ_0 de l'angle θ et commenter l'évolution de la période des oscillations. Était-ce prévisible ? Justifier.

APPEL PROF 1

Isochronisme ?

On s'intéresse maintenant à l'équation complète du pendule (Éq. 1).

5. En posant $x = \theta$ et $y = \omega$, exprimer $\dot{x}$ et $\dot{y}$ en fonction de x et y.
6. Compléter la fonction `pendule` associée à l'équation différentielle du pendule qui permettra de calculer numériquement $\theta(t)$ avec la fonction `odeint` (Doc. 1).
7. Représenter sur un même graphique l'évolution de l'angle θ dans le cas du pendule simple et de l'oscillateur harmonique pour $\theta_0 = 0{,}1$ et $\omega_0 = 0$.
8. Que remarque-t-on pour des oscillations d'amplitude plus importante ? Interpréter.
9. Comparer l'évolution de $\theta(t)$ pour $\theta_0 = \pi$ et $\omega_0 = 0{,}1$ dans le cas de l'oscillateur harmonique et du pendule. Interpréter.

APPEL PROF 2

10. On revient à $\omega_0 = 0$. Mesurer la période T des oscillations pour quelques valeurs de θ_0. Créer un nouveau graphe avec `plt.figure(2)` et représenter $T(\theta_0)$.
11. La formule de Borda permet d'approcher la période T du pendule en fonction de l'amplitude des oscillations θ_0 :

$$T(\theta_0) \approx T_0\left(1 + \frac{\theta_0^2}{16}\right), \text{ avec } T_0 = 2\pi\sqrt{\frac{\ell}{g}}.$$

Écrire la fonction `borda(theta0)` qui renvoie la période approchée du pendule simple en fonction de l'amplitude des oscillations. Représenter sur un même graphe les mesures réalisées (Q. 10) et les valeurs calculées avec cette formule. Commenter.

12. ❄ Le calcul exact est possible numériquement. La fonction `periode_int(theta0)` donne, pour une valeur de θ_0, la période des oscillations. Représenter graphiquement cette fonction. Commenter.

Énergie mécanique

En l'absence de frottement, le point M n'est soumis qu'à une force conservative (le poids) et une force qui ne travaille pas (la tension du fil) : c'est un mouvement conservatif. La vérification de la conservation de l'énergie lors de la résolution numérique permet donc de vérifier la pertinence des résultats obtenus.

13. Exprimer l'énergie cinétique $\mathcal{E}_\mathrm{c}$ du point matériel M en fonction de m, ℓ et ω.

14. De même, exprimer l'énergie potentielle $\mathcal{E}_\mathrm{p}$ du point M et en déduire son énergie mécanique $\mathcal{E}_\mathrm{m}$.

15. Écrire les fonctions

```
energie_cinetique(omega)
energie_potentielle(theta)
energie_mecanique(theta, omega)
```

associées à ces énergies. Dans une nouvelle fenêtre et sur le même graphe, représenter $\mathcal{E}_\mathrm{c}(t)$, $\mathcal{E}_\mathrm{p}(t)$ et $\mathcal{E}_\mathrm{m}(t)$. Commenter.

❄ **16.** La fonction `euler_deuxieme_ordre(f, V0, t)` intègre une équation différentielle du deuxième ordre en utilisant la méthode d'Euler. Elle s'utilise comme `odeint`. Commenter l'évolution de l'énergie mécanique dans le cas où l'équation du pendule est résolue avec la méthode d'Euler. Conclure.

❄ **17.** On revient à l'utilisation d'`odeint`. Modifier la fonction `pendule` pour prendre en compte des frottements fluides de la forme $\overrightarrow{f} = -\lambda\overrightarrow{v}$. Commenter alors l'évolution de $\mathcal{E}_\mathrm{m}(t)$.

Document 1 – Intégration d'équations différentielles avec Python

`scipy.integrate.odeint(F, V0, t)` : intègre un système d'équations différentielles du premier ordre de la forme :

$$\frac{\mathrm{d}V}{\mathrm{d}t} = F(V,t), \text{ où } V = \begin{pmatrix} x \\ \dot{x} \\ \vdots \end{pmatrix}.$$

Paramètres :

`F` : fonction qui calcule la dérivée de V en t ;

`V0` : conditions initiales sur V ;

`t` : liste des instants auxquels est calculé V.

Renvoie :

`V` : tableau contenant `len(t)` vecteurs V, calculés aux instants de `t`.