

Semaine 24 : du 23 au 27 mars

Au programme

Chapitre M7 : Mouvement conservatif à un degré de liberté

- ▶ Lecture d'un profil d'énergie potentielle : points de vitesse nulle, positions accessibles, positions d'équilibre stables et instables, état lié et état de diffusion ;
 - ▶ Caractérisation des positions d'équilibre par dérivation de l'énergie potentielle ;
 - ▶ Approximation harmonique au voisinage d'une position d'équilibre stable : développement limité de l'énergie potentielle, équation du mouvement linéarisée, pulsation des petites oscillations ;
 - ▶ Adimensionnalisation d'une équation différentielle, système d'unités naturelles, nombre de dimensions indépendantes (on ne peut pas tout adimensionner comme on veut !);
 - ▶ Méthode d'Euler pour une équation différentielle d'ordre 2, risque de divergence non-physique (constatée sur un exemple mais non démontrée) ;
 - ▶ Utilisation de `scipy.integrate.odeint` ;
 - ▶ Effet qualitatif des non-linéarités sur le mouvement : perte d'harmonicité et enrichissement spectral, perte d'isochronisme des oscillations, dissymétrie des oscillations par rapport à la position d'équilibre.
- ✖ J'ai travaillé sur l'exemple du potentiel de Morse, mais rien n'est à connaître à ce sujet ;
- ✖ Les spécifications de la fonction `odeint` ne sont pas à connaître et doivent être rappelées.

Chapitre T3 : Machines thermodynamiques

- ▶ Classification : moteur, machine frigorifique ou réceptrice ; machine à écoulement et machine à piston(s) ; fluide caloporteur ou frigorigène ;
 - ▶ Modèle de source de chaleur pour un processus complexe (réaction chimique, renouvellement du mélange gazeux post-combustion, etc.) ;
 - ▶ Notion de machine ditherme, intérêt (une seule source de chaleur à construire, l'autre étant l'environnement) ;
 - ▶ Diagramme des échanges d'un moteur et d'une machine frigorifique dithermes ;
 - ▶ Performance de la machine : rendement/efficacité/COP, ordres de grandeur ;
 - ▶ Cycle dans le diagramme de Watt : aire et travail échangé, sens de parcours et nature de la machine ;
 - ▶ Premier principe industriel (dont démonstration hors-programme).
- ✖ Le second principe n'a pas encore été étudié, et donc le théorème de Carnot non plus ;
- ✖ Les changements d'état n'ont pas encore été étudiés, les exemples sont donc restreints aux machines fonctionnant avec un gaz parfait.

Applications de cours

Ces applications de cours sont des « briques élémentaires » des raisonnements à mener dans les exercices : les maîtriser est incontournable. Elles sont toutes traitées de manière exhaustive dans le cours.

Le travail demandé consiste à se les approprier, afin d'être capable de les réinvestir dans un sujet d'écrit ou d'oral. Je n'attends pas des étudiants un apprentissage par cœur, mais j'attends qu'ils les aient travaillées suffisamment pour les mener à bien en autonomie, c'est-à-dire savoir refaire seul les raisonnements, sans aide de l'interrogateur.

M7.1 - Montrer à l'aide d'un développement limité que les petites oscillations au voisinage d'une position d'équilibre stable peuvent s'approximer par celles d'un oscillateur harmonique. Exprimer la pulsation des oscillations en fonction des caractéristiques du profil d'énergie potentielle. On raisonnera sur une énergie potentielle quelconque $E_p(x)$ dépendant d'une coordonnée cartésienne x .

M7.2 - Compléter le code ci-dessous pour résoudre l'équation différentielle adimensionnée $\ddot{x} = f(x)$ par la méthode d'Euler et représenter la courbe $x(t)$. On établira les relations de récurrence utiles avant toute écriture de code.

M7.3 - Compléter le code ci-dessous pour résoudre l'équation différentielle adimensionnée $\ddot{x} = f(x)$ en utilisant la fonction `odeint` du module `scipy.integrate` et représenter la courbe $x(t)$. On rappelle que cette fonction permet de résoudre un système différentiel écrit sous la forme

$$\frac{d\vec{y}}{dt} = F(\vec{y}, t).$$

La fonction `odeint(F, y0, t)`

- ▷ prend trois arguments obligatoires : la fonction d'évolution $F(y, t)$, une condition initiale y_0 et une liste ou tableau numpy d'instants t ;
- ▷ renvoie le vecteur Y à tout instant contenu dans t sous forme d'un tableau numpy à $n + 1$ dimensions.

On explicitera le vecteur $\vec{y}$ et la fonction d'évolution F avant toute écriture de code.

```

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

4
5 ### Paramètres de la simulation :
6 dt = 0.001          # pas de temps adimensionné
7 t_max = 70          # durée de la simulation adimensionnée
8 N = int(t_max/dt)   # nombre de points de la simulation
9
10
11 ### Conditions initiales :
12 x0 = 0              # pas d'énergie potentielle (fond du puits)
13 v0 = .5
14
15 ### Force adimensionnée dérivant du potentiel de Morse
16 def f(x):
17     return - np.exp(-x) * (1 - np.exp(-x))
18
19 ### code à écrire à partir d'ici.
```

T3.1 - Pour un type de machine au choix de l'interrogateur, représenter le diagramme des échanges, indiquer le sens/signé des échanges d'énergie et définir le rendement/efficacité. Montrer selon la machine étudiée si le rendement/efficacité est ≥ 1 . Donner un ordre de grandeur pour une machine réelle.

T3.2 - Un compresseur industriel aspire de l'air dans l'atmosphère ($T_0 = 20^\circ\text{C}$, $P_0 = 1\text{ bar}$) et le comprime jusqu'à $P_1 = 10\text{ bar}$ de manière adiabatique réversible. Représenter l'évolution sur un diagramme de Clapeyron. Déterminer la température T_1 en sortie du compresseur, puis le travail massique reçu par l'air.

À quoi s'attendre pour les programmes suivants ?

- ▷ Chapitre M8 : Forces centrales conservatives.
- ▷ Chapitre C7 : Diagrammes potentiel-pH.