TP 17 : Méthode d'Euler
Méthodes numériques

adapté de M. Melzani

Les points du programme :
- Capacité numérique : mettre en œuvre la méthode d'Euler à l'aide d'un langage de programmation pour simuler la réponse d'un système linéaire du premier ordre à une excitation de forme quelconque.
- Capacité numérique : à l'aide d'un langage de programmation, mettre en évidence le non isochronisme des oscillations pour un pendule.

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 1.	Mettre en œuvre la méthode d'Euler explicite afin de résoudre une équation différentielle d'ordre 1.
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).

Objectifs

- Comprendre et savoir mettre en œuvre la méthode d'Euler explicite.

Matériel : *Votre ordinateur avec Anaconda (Spyder).*

La méthode d'Euler est une méthode numérique qui permet de ***résoudre numériquement une équation différentielle*** en approximant les dérivées par des développements limités.

Idées de l'algorithme

1. On utilise une approximation pour la dérivée. Il y a plusieurs possibilités. La ***méthode d'Euler explicite*** utilise l'approximation suivante :

$$\frac{dX(t)}{dt} \approx \frac{X(t+dt)-X(t)}{dt}$$ avec dt une durée très courte.

L'équation différentielle du 1er ordre $\frac{dX}{dt}+\frac{1}{\tau}X=\frac{X_\infty}{\tau}$ devient alors

$$\frac{X(t+dt)-X(t)}{dt}+\frac{1}{\tau}.X(t)=\frac{X_\infty}{\tau} \quad \Longrightarrow \quad \boxed{X(t+dt)=X(t)+\underbrace{\frac{X_\infty-X(t)}{\tau}}_{F}.dt}$$

Cette dernière expression permet de calculer $X(t+dt)$ si on connaît $X(t)$.

2. On va donc procéder successivement :
- On part de X_0 qui est $X(t=0)$, on le met dans une liste `liste_X`.
- On en déduit $X(dt)=X_0+F.dt$ et on le met dans la liste.
- On en déduit $X(2.dt)=X(dt)+F.dt$ et on le met dans la liste.
- Etc... on en déduit la valeur de X à tous les instants $i.dt$, et on s'arrête quand cela suffit.

Remarque : Le paramètre F peut dépendre comme ici de la valeur de $X(t)$.

Retour sur les systèmes du 1er ordre
Chute avec frottements fluides ($en-k.v^2$)

Lors d'un saut en parachute, l'équation différentielle réagissant l'évolution de la vitesse du parachutiste (avant ouverture du parachute) est la suivante (voir DM2) :

$$\frac{dv(t)}{dt}+\frac{k}{m}.v^2(t)=g$$

On prendra comme condition initiale $v(t=0)=0$. Remarque (axe vertical orienté vers le bas pour avoir une vitesse positive)

Q1. Cette équation différentielle peut-elle être facilement résolue ? Pourquoi ?

Nous allons donc procéder à une résolution numérique grâce à la méthode d'Euler explicite (voir ci-contre pour des rappels).

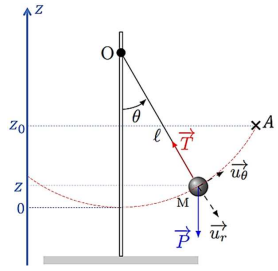
- ☐ Récupérer sur le cahier de prépa le fichier *TP17_Chute.py*. L'ouvrir avec Spyder.
- ☐ Compléter le script afin de mettre en œuvre la méthode d'Euler.
- ☐ Compléter le script afin d'afficher l'évolution de la vitesse en fonction du temps.
- ☐ Exécuter le script et vérifier que le résultat est cohérent avec celui attendu (existence d'une vitesse limite).

Q2. Exprimer la position $z(t)$ du parachutiste en fonction de la vitesse $v(t)$.

Q3. Réécrire cette relation en utilisant l'approximation de la dérivée utilisée dans la méthode d'Euler explicite.

- ☐ Ajouter des lignes de codes dans la méthode d'Euler pour obtenir la liste, appelée `liste_z`, des positions successives du parachutiste. On prendra comme condition initiale $z(0)=0$.
- ☐ Ajouter une deuxième figure afin d'afficher l'évolution de la position z en fonction du temps.

Systèmes du 2ème ordre
Pendule simple



On souhaite maintenant 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**. La méthode d'Euler n'est pas applicable directement ici.

➔ 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 :

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

Avec l'approximation de la dérivée on peut écrire :

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

Ce système d'équations du 1er ordre permet donc de calculer $\theta(t+dt)$ et $\dot{\theta}(t+dt)$ si on connaît $\theta(t)$ et $\dot{\theta}(t)$.

- ☐ Récupérer sur le cahier de prépa le fichier *TP17_Pendule.py*. L'ouvrir avec Spyder.
- ☐ Compléter le script afin de mettre en œuvre la méthode d'Euler.
- ☐ Compléter le script afin d'afficher l'évolution de l'angle θ en fonction du temps.
- ☐ Exécuter le script et vérifier que le résultat est cohérent avec celui attendu pour un angle faible.
- ☐ Faire plusieurs essais en modifiant l'angle de départ.

Q4. Jusqu'à quelle valeur de l'angle de départ peut-on considérer que le pendule se comporte comme un oscillateur harmonique (période indépendante de l'angle de départ, aussi appelée ***isochronisme***) ?

Utilisation de la fonction odeint

Il est possible de résoudre plus 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 $y_0(t)$ (donc dans notre exemple, de $\theta(t)$), et la seconde colonne contient les valeurs de $y_1(t)$ (donc dans notre exemple, de $\dot{\theta}(t)$).

- ☐ Récupérer sur le cahier de prépa le fichier *TP17_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 identique à celui obtenu précédemment avec la méthode d'Euler.

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.

3ème exemple : Transitoire d'un RLC série (pour les plus rapides)

- ☐ Récupérer sur le cahier de prépa le fichier *TP17_RLC.py*. L'ouvrir avec Spyder.
- ☐ Compléter dans le script les expressions de la pulsation propre du circuit et du facteur de qualité.
- ☐ Compléter la fonction F afin qu'elle décrive le comportement de la tension u_c aux bornes du condensateur.
- ☐ Compléter le script afin d'afficher l'évolution temporelle de u_c.
- ☐ Exécuter le script et vérifier que le résultat est cohérent le comportement du RLC série.