SIMULATIONS NUMÉRIQUES : ÉQUATIONS DIFFÉRENTIELLES DE DIMENSION 1

1 Introduction

1.1 Objectifs du Tp

L'objectif de ce tp est de traiter un système d'équations différentielles de dimension N pouvant se mettre sous la forme d'une équation différentielle du premier ordre d'une fonction **Y** de dimension N.

Les schémas d'intégrations utilisés suivant seront comparés à la solution données par la fonction `odeint` du module `scipy.integrate` :

- méthode d'Euler explicite
- méthode de Heun
- méthode de Runge Kunta 4

Les problèmes suivants servent de support :

- Cinétiques chimiques simples A donne B donne C
- Système masse/ressort amorti
- Pendule simple non linéarisé

1.2 Rappels

Résoudre un problème de Cauchy consiste à trouver la fonction **Y** de $[t_0, t_f] \rightarrow \mathbb{R}^N$, telle que :

$$\begin{cases} \dfrac{d\mathbf{Y}}{dt} = \mathbf{F}(\mathbf{Y}, t) \\ \mathbf{Y}(t_0) = \mathbf{Y_0} \end{cases} \quad \text{où } t \in [t_0, t_f] \text{ et } \mathbf{Y_0} \in \mathbb{R}^N \tag{1}$$

2 Système d'équations différentielles du premier ordre

2.1 Problèmes

On considère deux problèmes pouvant se ramener à un système d'équations différentielles du premier ordre :

- succession de deux réactions chimiques du premier ordre : $A \xrightarrow{k_1} B \xrightarrow{k_2} C$
- un système masse ressort écarté de sa position d'équilibre x_0 : $m.\dfrac{d^2x(t)}{dt^2} = -k.[x(t) - x_0] - \lambda.\dfrac{dx(t)}{dt}$

2.2 Mise en équations

Le problème de cinétique chimique se ramène à :

$$\begin{cases} \dfrac{\mathrm{dA}(t)}{\mathrm{dt}} & = & -k_1.\mathrm{A}(t) \\ \dfrac{\mathrm{dB}(t)}{\mathrm{dt}} & = & k_1.\mathrm{A}(t) - k_2.\mathrm{B}(t) \\ \dfrac{\mathrm{dC}(t)}{\mathrm{dt}} & = & k_2.\mathrm{B}(t) \end{cases}$$

avec $\mathrm{A}(0) = 0,1$ mol/L, $\mathrm{B}(0) = 0$ mol/L et $\mathrm{C}(0) = 0$ mol/L

Le problème masse-ressort se ramène à :

$$\begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{dt}} & = & v(t) \\ \dfrac{\mathrm{d}v(t)}{\mathrm{dt}} & = & -\omega_0^2.[x(t) - x_0] - 2.\xi.\omega_0.v(t) \end{cases}$$

avec $\omega_0 = \sqrt{km}$ et $\xi = \lambda 2.\sqrt{k.m}$ pour les constantes et $x_0 = 0,1$ m, $x(0) = x_1 = 0,12$ m et $v(0) = 0$ m/s.

Pour revenir au problème générale (1), on pose pour le problème de cinétique chimique :

$$\mathbf{Y}(t) = \begin{bmatrix} y_0(t) \\ y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \mathrm{A}(t) \\ \mathrm{B}(t) \\ \mathrm{C}(t) \end{bmatrix} \quad \text{avec} \quad \mathbf{Y}(t_0) = \mathbf{Y_0} = \begin{bmatrix} 0,1 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbf{F}(\mathbf{Y}, t) = \begin{bmatrix} -k_1.y_0(t) \\ k_1.y_0(t) - k_2.y_1(t) \\ k_2.y_1(t) \end{bmatrix}$$

Concernant le système masse-ressort, on pose :

$$\mathbf{Y}(t) = \begin{bmatrix} y_0(t) \\ y_1(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} \quad \text{avec} \quad \mathbf{Y}(t_0) = \mathbf{Y_0} = \begin{bmatrix} x_1 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbf{F}(\mathbf{Y}, t) = \begin{bmatrix} y_1(t) \\ \omega_0.(x_0 - y_0(t)) - 2.\xi.\omega_0.y_1(t) \end{bmatrix}$$

2.3 Résolution numérique

Q - 1 : *Construire les fonctions* `Fchimie(Y,t)` *et* `Fpfd(Y,t)` *associées aux problèmes de cinétique chimique et au système masse-ressort. Les paramètres k_1, k_2, x_0, ξ et ω_0 seront associés à des variables globales.*

Comme dans le cas à une dimension, le schéma d'Euler explicite permet de passer de l'état i à l'état $i+1$ grâce à la relation de récurrence : $\mathbf{Y_{i+1}} = \mathbf{Y_i} + h.\mathbf{F}(\mathbf{Y_i}, t_i)$, où $h = t_{i+1} - t_i$. Ainsi au temps t_{i+1} :

- si `Y` est un tableau, la récurrence en **Python** donne pour $\mathbf{Y_{i+1}}$:

```
Y[i+1]=Y[i]+(T[i+1]-T[i])*F(Y[i],T[i])
```

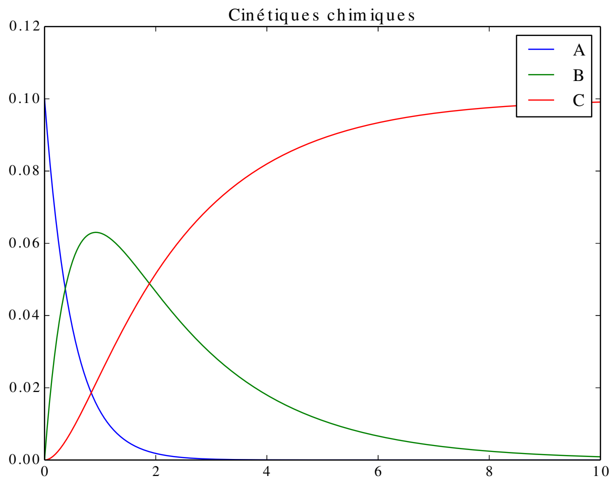
- si `Y` est une liste de listes, la récurrence en **Python** donne pour chaque composante j de $\mathbf{Y_{i+1}}$:

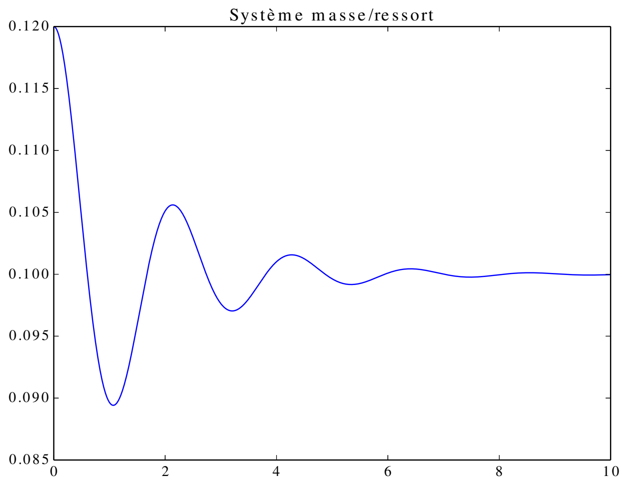
```
Y[i+1][j]=Y[i][j]+(T[i+1]-T[i])*F(Y[i],T[i])[j]
```

Q - 2 : *Construire, en utilisant un schéma d'Euler explicite, la fonction* `EulerExpDimN(F,Y0,T)` *qui permet de trouver l'évolution de* $\mathbf{Y}(t)$ *à partir de l'état initial* $\mathbf{Y_0}$ *pour toutes les valeurs de t contenues dans* `T`*. La fonction doit s'adapter automatiquement à la longueur de* `Y0` *dans les 2 cas.*

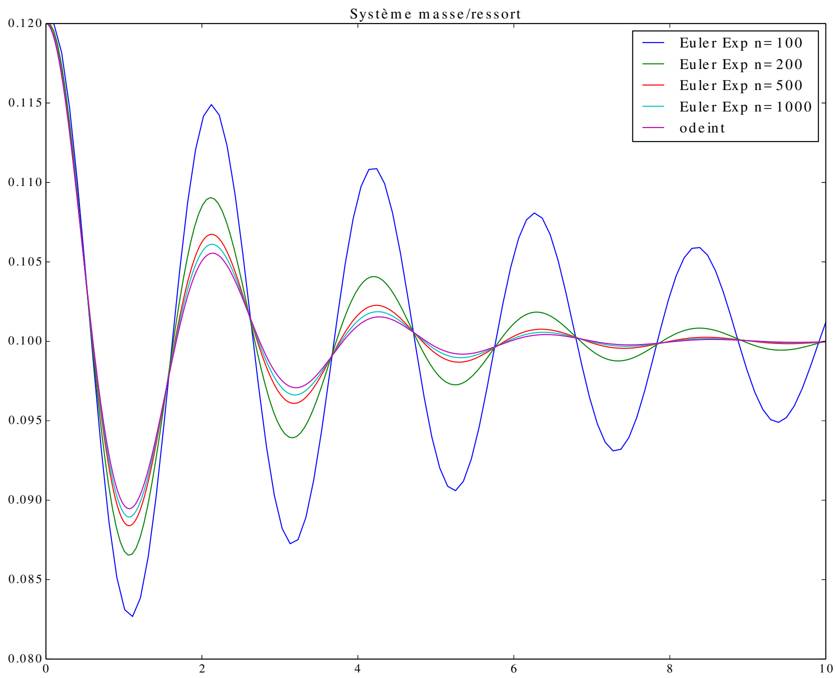
On sollicite les deux problèmes pour $t \in [0,10]$. On prend comme paramètres $(k_1;k_2) = (2;0,5)$ et $(x_0;\omega_0;\xi) = (0,1;3;0,2)$.

Q - 3 : *Appliquer la fonction* `EulerExpDimN` *aux problèmes de chimie et masse-ressort pour déterminer l'évolution des systèmes.*



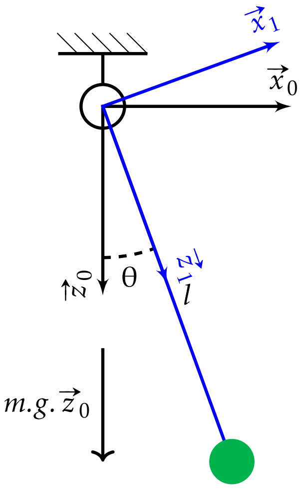


Q - 4 : *Étudier l'influence du nombre de points (ou de la longueur du pas temps).*



3 Pendule simple

3.1 Problématique



Objectifs :

- Déterminer l'évolution au cours du temps du mouvement du pendule simple
- Comparer cette évolution dans le cas où l'équation de mouvement est linéarisée avec celui où elle ne l'est pas.

Le cours de physique permet d'établir qu'au cours du temps :

$$m.l.\ddot{\theta} + m.g.\sin(\theta) = 0$$

On considère alors le problème suivant : Trouver θ de $\mathbb{R}^+ \to \mathbb{R}$ telle que

$$\ddot{\theta}(t) + \omega^2.\sin(\theta(t)) = 0 \quad \text{avec} \quad \theta(0) = \theta_0 \quad ; \quad \dot{\theta}(0) = \Omega_0 \quad \text{et} \quad \omega = \sqrt{gl} \tag{2}$$

Pour l'étude, nous prendrons $t \in [0, 2.\pi]$, $\omega = 2$ rad/s, $\theta_0 = 20°$ et $\Omega_0 = 0$

3.2 Solution harmonique

3.2.1 Equation différentielle linéaire à coefficients constants

Dans le cas des petites oscillations du pendule, il est possible d'approximer $\sin(\theta)$ par θ à l'aide d'un développement limité à l'ordre 1. Le problème posé en (2) devient alors:

$$\ddot{\theta} + \omega^2.\theta = 0 \quad \text{avec} \quad \theta(0) = \theta_0 \quad ; \quad \dot{\theta}(0) = \Omega_0 \quad \text{et} \quad \omega = \sqrt{gl} \tag{3}$$

admettant alors comme solution générale de l'équation sans second membre $\theta(t) = \lambda.\cos(\omega.t) + \mu.\sin(\omega.t)$. Avec les conditions initiales imposées :

$$\begin{cases} \theta(0) &= \lambda = \theta_0 \\ \dot{\theta}(0) &= \mu.\omega = \Omega_0 \end{cases} \Rightarrow \quad \theta(t) = \theta_0.\cos(\omega.t) + \Omega_0\omega.\sin(\omega.t)$$

3.2.2 Problème de Cauchy

Q - 5 : *Transformer le problème précédent en un problème de Cauchy.*

On utilise alors un vecteur de dimension 2 : $\mathbf{Y} = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}$

Q - 6 : *Adapter la méthode d'Euler explicite programmée dans le tp précédent pour résoudre le problème (3).*

3.2.3 Récurrence directe

En calculant la dérivée seconde à partir des formes discrétisées des dérivées premières, on obtient :

$\dot{y}(t_i) \approx Y_{i+1} - Y_i h \quad \text{et} \quad \ddot{y}(t_i) \approx \dot{y}(t_{i+1}) - \dot{y}(t_i) h \quad \Rightarrow \quad \ddot{y}(t_i) \approx Y_{i+2} - 2.Y_{i+1} + Y_i h^2$

Q - 7 : *Ecrire la relation de récurrence permettant d'obtenir* Y_{i+2} *en fonction de* Y_{i+1} *et* Y_i.

Q - 8 : *Ecrire un programme permettant de tracer l'évolution de* $\theta(t)$ *sur* $[0, 2.\pi]$ *avec la relation de récurrence précédente.*

3.2.4 Bibliothèque Python

La résolution numérique des équations différentielles est implantée dans **Python** . Il s'agit de la fonction **odeint**, de la bibliothèque **scipy.integrate**. Il convient donc de la charger :

```
from scipy.integrate import odeint
odeint(f,y0,X)
```

Comme dans le Tp précédent, les arguments sont :

- la fonction f
- le vecteur de valeurs initiales $y0$
- Une discrétisation X de l'intervalle sur lequel est intégrée l'équation différentielle

Q - 9 : *Comparer alors les différentes méthodes.*

3.3 Cas des grands angles

Dans l'hypothèse où les angles sont grands, l'équation différentielle ne peut pas être linéarisée.

Q - 10 : *Reprendre les questions de la partie précédente avec le problème défini par (2).*

Q - 11 : *Comparer l'erreur commise par la linéarisation de l'équation différentielle pour différentes valeurs de* θ_0 *pour* $\theta_0 \in [0, 90]$.