TP Info 14 – Résolutions numériques d'équations différentielles

D.Malka – MPSI 2014-2015 – Lycée Saint-Exupéry

A REMPLACER PAR DETERMINATION DE LA PORTEE MAX POUR UN TIR DE PROJECTILE AVEC FROTTEMENTS QUADRATIQUE!

SAUT FELIX BAUMGARTNER AVEC MODELISATION EXP DE LA DENSITE + coeff de trainee donnee.

ETUDE DE STABILITE DE LA SOLUTION : ON TRACE DIFFERENT GRAPHE POUR DIFFERENTES C.I.S

I1 – Méthode de Runge-Kutta d'ordre 4

On cherche à résoudre, sur $[a,b]$ une équation différentielle du type :

$$\begin{cases} y' = F(t, y(t)) \\ y(a) = y_0 \end{cases}$$

La méthode Runge-Kutta permet de résoudre numériquement une équation différentielle c'est-à-dire d'évaluer y_{k+1} en fonction de y_k. La relation de récurrence, pour un pas de discrétisation h, est :

$$y_{k+1} = y_k + \frac{h}{6}(F(t_k, y_k) + 2F(t_k + \frac{h}{2}, \alpha_k) + 2F(t_k + \frac{h}{2}, \beta_k) + F(t_{k+1}, \gamma_k))$$

avec

$$\begin{aligned} \alpha_k &= y_k + \frac{h}{2}F(t_k, y_k) \\ \beta_k &= y_k + \frac{h}{2}F(t_k + \tfrac{h}{2}, \alpha_k) \\ \gamma_k &= y_k + hF(t_k + \tfrac{h}{2}, \beta_k) \end{aligned}$$

1. Programmer la méthode Runge-Kutta en langage Python.
2. Tester cette méthode sur la fonction exponentielle sur l'intervalle $[0,1]$ avec un pas $h = 1/3$.
3. Comparer à la méthode d'Euler.

I2 – Contrôle cinétique vs contrôle thermodynamique

Un produit A est susceptible de former un produit B ou un produit C suivant les deux réactions parallèles suivantes :

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} B \quad (1)$$

$$A \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} C \quad (2)$$

On se demande qui du produit B ou C est majoritaire dans la solution.

Chaque réaction obéit à la règle de Van't Hoff c'est-à-dire que les lois de vitesses sont données par :

$$\begin{aligned} v_1 &= k_1[A] \\ v_{-1} &= k_{-1}[B] \\ v_2 &= k_2[A] \\ v_{-2} &= k_{-2}[C] \end{aligned}$$

Les concentrations initiales sont nulles sauf pour A ($[A_0] = 1\, mol.L^{-1}$).

1. A l'équilibre les équilibres chimiques (1) et (2) vérifient respectivement $v_1 = v_{-1}$ et $v_2 = v_{-2}$. En déduire les expression des constantes d'équilibre K_1 et K_2 en fonction des constantes de vitesse.
2. Exprimer chacune des variations de concentrations $[A](t)$, $[B](t)$ et $[C](t)$ en fonction de v_1, v_{-1}, v_2 et v_{-2}.
3. En déduire le système d'équations vérifié par $[A](t)$, $[B](t)$ et $[C](t)$.
4. Résoudre ce système à l'aide des fonctions prédéfinies de Python pour $k_1 = 1$, $k_{-1} = 0.125$, $k_2 = 10$ et $k_{-2} = 2$.
5. Représenter $[A](t)$, $[B](t)$ et $[C](t)$ sur un même graphe. Répondre à la question initiale et interpréter à la lumière des constantes d'équilibre et des constantes de vitesse.

I3 – Parabole de sureté

On considère le problème de la chute libre d'un corps M de masse m dans le champ de pesanteur terrestre $\vec{g}$. On admet que la trajectoire est plane. La position initiale est $(0,0)$, la vitesse initiale $(v_0\cos\alpha, v_0\sin\alpha)$. On donne le programme qui calcule la solution du problème.

```
import numpy as np
import scipy.integrate as integ
import matplotlib.pyplot as plt

#PARAMETRES
m=1
g=9.8
v0=10
alpha=np.pi/6

def equation(pos,t):

    x=pos[0]
    dx=pos[1] #Vx
    y=pos[2]
    dy=pos[3] #Vx

    if y>=0:
        return[dx,0,dy,-g]


t=np.linspace(0,10,10000)
init=[0,v0*np.cos(alpha),0,np.sin(alpha)]

sol=integ.odeint(equation,init,t)
```

1. Représenter une quarantaine de parabole (uniquement pour $y >= 0$).
2. Ajouter la parabole d'équation $y = \dfrac{v_0^2}{2g} - \dfrac{g}{2v_0^2}x^2$ dite « parabole de sureté ». Commenter.
3. Modifier les équations précédentes pour introduire une force de frottement fluide $\vec{f} = -\lambda\vec{v}$. Tracer à nouveau quarante paraboles. Commenter.

I4 – Méthode balistique

Déterminer (avec 5 décimales significatives), une valeur de α telle que la solution de l'équation $y'' = 1 + y^3$ avec $y(0) = 1$ et $y'(0) = \alpha$ vérifie $y(1) = 1$.

On pourra faire appel à une méthode dichotomique au sein de la résolution.

I5 – Modèle Proie-Prédateur de Lotka-Volterra

Soit $P(t)$ la population d'un petit poisson et $R(t)$ la population de son prédateur. La dynamique de cette population pet-être décrite par le système d'équation ci-dessous.

$$\begin{cases} P'(t) = n_P P(t) - aR(t)P(t) \\ R'(t) = -m_R R(t) + bP(t)R(t) \end{cases}$$

1. Résoudre numériquement ce système à l'aide des fonction prédéfinies de Python pour $n_P = 10$, $m_R = 5$, $a = 0.01$ et $b = 0.001$.
2. Sur un même graphe, représenter $P(t)$ et $R(t)$. Commenter.
3. Si l'homme vient à pêcher les petits poissons, d'après vous, quelle population sera-t-elle la plus affectée ?
4. Tester votre hypothèse en modifiant les équations afin de tenir compte de la pêche.