TP CN1 – Résolutions numériques d'équations

D.Malka – MPSI 2019-2020 – Lycée Jeanne d'Albret

I1 – Méthode de la sécante (ou de Lagrange)

Soit $f : I \to \mathbb{R}$, s'annulant en $a \in I$, concave ou convexe au voisinage de a. On fixe u_0, $u_1 \in I$ au voisinage de a et on construit $(u_n)_{n\geq 2}$ de la façon suivante : pour tout $n \geq 2$, on définit u_n comme le zéro de la sécante passant par les points d'abscisses u_{n-1} et u_{n-2} à la courbe représentative de f.

1. Construire un schéma illustrant la méthode.
2. Donner la relation liant u_{n-2}, u_{n-1} et u_n.
3. Proposer un test d'arrêt pour la méthode de la sécante.
4. Programmer la méthode et l'utiliser par évaluer la valeur de $\sqrt{3}$ au ε-machine près.

☞ *Cette méthode, similaire à la méthode de Newton, admet pour ordre de convergence le nombre d'or* $\varphi = \frac{1+\sqrt{5}}{2}$ *! Elle converge donc un peu moins vite que la méthode de Newton (ordre 2).*

I2 – Point de fonctionnement d'un circuit à diode

On considère un circuit série constitué d'un générateur de force électromotrice égale à $E = 10\,\mathrm{V}$, de résistance interne $r = 50\,\Omega$ monté en série avec une diode dont la loi de fonctionnement est :

$$i_d(u) = I_s(e^{\frac{u}{V_t}} - 1)$$

avec $I_S = 1 \times 10^{-14}\,\mathrm{A}$ et $V_T = 25\,\mathrm{mV}$.

La loi de fonctionnement du générateur est :

$$i_g(u) = \frac{E - u}{r}$$

La diode et le générateur étant montés en série $i_g(u) = i_d(u)$.

1. Tracer la caractéristique de la diode.
2. Écrire une fonction `rech_zero` qui détermine le ou un zéro d'une fonction f passée en argument avec une erreur estimée égale à ε également passée en argument.
3. A l'aide de la fonction `rech_zero`, déterminer numériquement le point de fonctionnement du circuit.

I3 – Portée d'un tir de projectile avec frottement linéaire

On considère sait que la portée d'un projectile de masse m envoyé avec un vitesse $\vec{v}_0$, est maximale, à v_0 constant, pour un angle de tir $\alpha = 45\,^\circ$.

On se demande si ce résultat reste vrai en présence d'une force de frottement linéaire du type $-\lambda\vec{v}$.

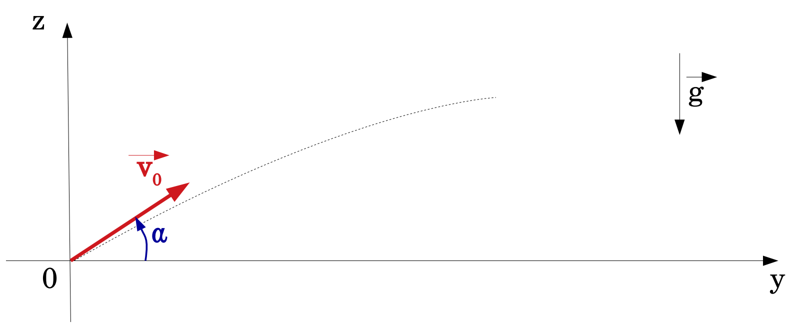

Figure 1 – Quelle est la portée d'un tir de projectile avec frottement ?

La résolution des équations du mouvement donne l'équation de la trajectoire suivante :

$$z(y) = (g\tau^2 + v_0\tau\sin\alpha)\frac{y}{y_{\text{inf}}} - g\tau^2\ln\left(\frac{y_{\text{inf}}}{y_{\text{inf}} - y}\right)$$

avec $\tau = \frac{m}{\lambda}$ et $y_{\text{inf}} = v_0\tau\cos\alpha$. y_{inf}, appelé « mur aérodynamique », est une borne supérieure de y.

Valeurs numériques proposées pour les applications : $v_0 = 5\,\mathrm{m\cdot s^{-1}}$, $\lambda = 0{,}5\,\mathrm{N\cdot m^{-1}\cdot s}$ et $m = 1\,\mathrm{kg}$.

Données : on pourra utiliser les fonctions `bisect` et `newton` du module `scipy.optimize`.

1. Tracer la trajectoire du projectile à l'aide des modules `matplotlib.pyplot` et `numpy` ou bien du module `pylab`. On utlisera les fonction `xlim` et `ylim` pour centrer le graphe sur la zone d'importance physique.
2. Portée du tir.
   - 2.1 Proposer une méthode numérique permettant de déterminer la portée du tir.
   - 2.2 Écrire une fonction `portee` qui renvoie la portée du tir de projectile suivant des conditions initiales données. Cette fonction prendra en argument les valeurs de m, λ, v_0, α ainsi que la précision e attendue sur la portée calculée.
   - 2.3 Tester cette fonction sur un cas particulier.
3. Angle de portée maximale.
   - 3.1 Écrire une fonction `maxi(l)` prenant en argument une liste `l` d'entiers ou de flottants et renvoyant la valeur maximale de cette liste ainsi que l'indice de cette valeur.
   - 3.2 Écrire une fonction `portee_max` qui estime l'angle α_m donnant une portée maximale au tir. Les paramètres m, λ, v_0 seront passés en argument de la fonction.
4. Écrire un programme permettant d'étudier l'influence de v_0 sur la valeur de l'angle de tir α_m donnant une portée maximale. Le résultat de cette étude prendra la forme d'un graphe représentant α_m en fonction de v_0.

I4 – Recherche du minimum d'une fonction

Le but de cet exercice est de proposer un algorithme inspiré de l'algorithme de dichotomie pour approcher le minimum d'une fonction convexe sur un segment de $\mathbb{R}$.

1. **Dichotomie**
   - 1.1 Écrire en langage Python la fonction `bisect` avec les arguments adéquats déterminant à e près le zéro d'une fonction f dont on sait qu'il appartient à l'intervalle $[a, b]$.
   - 1.2 Montrer la terminaison de la fonction `bisect`
   - 1.3 Montrer la correction de la fonction `bisect`.
2. **Minimum d'une fonction convexe**

   On dit qu'une fonction f est convexe sur un intervalle I, lorsque tout segment reliant deux points de sa courbe représentative se trouve au dessus de la courbe. On admettra que si une fonction est convexe sur $I = [a, b]$, elle y admet un minimum atteint en un seul point que l'on notera μ. Notre objectif est de décrire et d'implémenter un algorithme de calcul approché de μ.

   Notations : pour x_1, x_2 tels que $a < x_1 < x_2 < b$, on note p_0, p_1, p_2 les pentes des sécantes à la courbe représentative de f aux points d'abscisses (a, x_1), (x_1, x_2) et (x_2, b).

   - 2.1 A l'aide d'une illustration graphique dire si chacune des conjectures suivantes vous paraît raisonnable :
     — $p_0 \leq p_1 \leq p_2 \leq 0 \Rightarrow \mu \in [x_2, b]$;
     — $p_0 \leq p_1 \leq 0 \leq p_2 \Rightarrow \mu \in [x_1, b]$;
     — $p_0 \leq 0 \leq p_1 \leq p_2 \Rightarrow \mu \in [a, x_2]$;
     — $0 \leq p_0 \leq p_1 \leq p_2 \Rightarrow \mu \in [a, x_1]$.
     Y-a-t-il a priori d'autres cas possibles ? Justifier.
   - 2.2 En vous inspirant de la méthode de dichotomie, écrire en langage Python la fonction `min_function` calculant de façon approchée (à e près) le minimum de f dont on sait qu'il existe sur un intervalle $[a, b]$. Pour cela on posera :

$$\begin{cases} x_1 = m - \lambda(b-a) \\ x_2 = m + \lambda(b-a) \end{cases} \text{avec } m = \frac{a+b}{2} \quad \text{et} \quad 0 < \lambda < 1/2$$

   - 2.3 Exprimer la complexité, dans le pire des cas, de la fonction `min_function` en fonction de ε, b, a et λ. Estimer alors le nombre d'itérations pour $\lambda = 1/3$, $e = 1\times10^{-6}$ et $b - a = 1$.