TP CN1 – Résolutions numériques d'équations

D.Malka – MPSI 2018-2019 – 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 e 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 $-\lambda\vec{v}$.

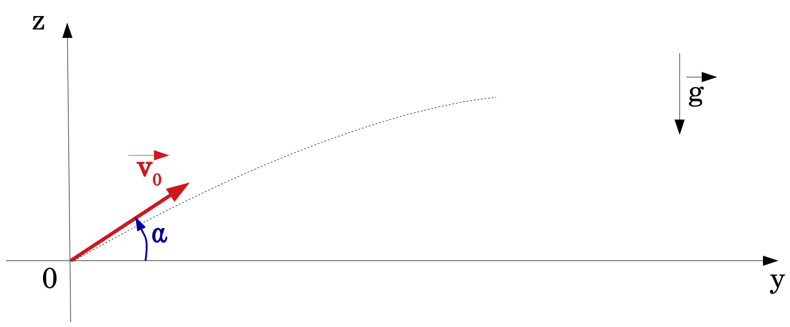

Figure 1 – Quelle est la portée d'un tir de projectile avec frottement ?

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 $m = 1\,\mathrm{kg}$.

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.

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

I4 – Convergence des méthodes numériques

1. Rechercher un zéro de la fonction sinus sur $[-\frac{\pi}{2}, \frac{\pi}{2}]$ avec la méthode de Newton (en initialisant à $\pi/3$) puis avec la méthode dichotomique. Comparer les vitesses de convergence avec la fonction `time()` du module `time()`.
2. Même question en initialisant la méthode de Newton à $\pi/2$.
3. Tester la méthode de Newton sur la fonction $f : x \to \sqrt{x}$. Expliquer le résultat. La méthode dichotomique est-elle applicable ?
4. Combien d'itérations sont-elles nécessaires pour obtenir p bits significatifs avec la méthode dichotomique ? Avec la méthode de Newton ?