MNI-TP1-2014

March 12, 2015

TP 1 – Méthodes Numériques pour l'Ingénieur CM3 – Mars 2015

1 Résolution des équations non linéaires

Le texte de cette sessions de travaux pratiques est également disponible ici
http://nbviewer.ipython.org/github/ecalzavarini/numerical-methods-at-polytech-lille/blob/master/MNI-TP1-2014.ipynb

1.0.1 Instructions pour ce TP

Pendant ce TP vous aurez à écrire plusieurs scripts (Nous vous suggérons de les nommer script1.py , script2.py ,. . .)

Les scripts doivent être accompagnés par un par un document descriptif unique (README.txt). Dans ce fichier, vous devrez décrire le mode de fonctionnement des scripts et, si besoin, mettre vos commentaires. Merci d'y écrires aussi vos nomes et prenoms complets.

Tous les fichiers doivent etre mis dans un dossier appelé TP1-nom1-nom2 et ensuite être compressés dans un fichier TP1-nom1-nom2.tgz .

Enfin vous allez envoyer ce fichier par email à l'enseignant :
soit Enrico (enrico.calzavarini@polytech-lille.fr) ou Stefano (stefano.berti@polytech-lille.fr)

1.1 Programmation

Ecrire un script Python permettant la recherche des racines d'une équation quelconque

$$f(x) = 0$$

par la méthode :
a) de la dichotomie,
b) de la tangente (de Newton-Raphson) ,
en appliquant le critère d'arrêt $x_{n+1} - x_n < \varepsilon_1$, où la valeur de ε_1 sera précisée par l'utilisateur.
L'équation $f(x)$ sera définie à l'aide d'une function dans le script.

1.2 Validation

Déterminer la racine de l'équation

$$f(x) = x^4 - 2x^3 - 11x^2 + 12x$$

par la méthode dichotomique (dans l'intervalle $x \in [3.2, 8.2]$) et de la tangente (en commençant les itérations par $x_0 = 8.2$).

On effectuera les calculs de la racine avec les précisions $\varepsilon_1 = 10^{-k}$ avec $k = 1, \ldots, 6$.

Tracer au préalable le graphique de la fonction $f(x)$ dans l'intervalle étudié.

Pour chaque calcul préciser : la valeur de la racine x trouvée, de la fonction $f(x)$, de l'erreur absolue $\varepsilon = x - x^*$ (avec x^* le valeur exacte, égale ici à 4.0) ainsi que le nombre d'itérations effectuées.

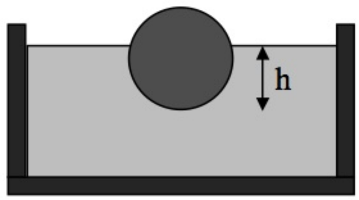
Les valeurs obtenues doivent être affichés de façon claire sur l'écran par le script et dans le compte-rendu dans un tableau.

Si possible, formuler une conclusion sur les résultats obtenus.

1.3 Application

Une boule de rayon R et de masse volumique ρ_l est placée dans un réservoir rempli d'un liquide au repos de masse volumique $\rho_f = 1000Kg/m^3$ (eau). La boule s'enfonce alors d'une hauteur h (voir la figure ci-dessous).

Le but de ce TP est de déterminer, à l'aide des deux méthodes numériques vues auparavant, cette hauteur h en fonction de la masse volumique de la boule ρ_l .



Question 1 : En considérant l'équilibre des forces en présence, donner l'expression permettant de déterminer $h = f(\rho_l, \rho_f, R)$.

Rappel :

Le volume d'une sphère du rayon r est donné par l'équation

$$V_s = 4/3\pi r^3$$

Le volume d'une calotte sphérique du rayon r et de hauteur h est

$$V_c = 1/3\pi h^2(3r - h)$$

Question 2 : Pour $R = 0.125m$, et pour les valeurs suivantes de ρ_l

a) $\rho_l = 1800Kg/m^3$ (plexiglas) ,

b) $\rho_l = 1000Kg/m^3$ (caoutchouc) ,

c) $\rho_l = 400Kg/m^3$ (pin) ,

la boule coulera-t-elle ou non? Sinon, de quelle profondeur h s'enfoncera-t-elle?

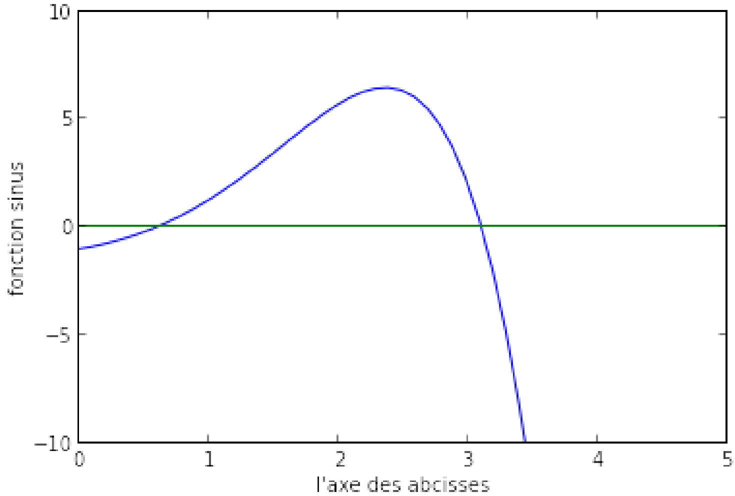
Tracer pour le trois cas le graphique de la fonction $f(x)$ dans l'intervalle étudié.

Bonus: Déterminer $h = f(\rho_l, \rho_f, R)$ à l'aide de la méthode de la corde (de la regula-falsi).

Rappel : pour tracer un graphique en Python

```
In [26]: import numpy as np
         import matplotlib.pyplot as plt

         x=np.linspace(-5,5,100)  # nous définissons une liste (array) avec Numpy
         plt.plot(x,np.sin(x)*np.exp(x)-1)  # on utilise la fonction sinus de Numpy
         plt.plot(x,np.zeros(100))
         plt.ylabel('fonction sinus')
         plt.xlabel("l'axe des abcisses")
         plt.axis([0, 5, -10, 10])
         plt.show()
```



10
5
0
−5
−10
0
1
2
3
4
5
fonction sinus
l'axe des abcisses