

© J.-B. A. K. <jean-baptiste.apoung@math.u-psud.fr>

Fiche : préparation au TP 3

Cette fiche porte sur la méthode du point fixe. Elle introduit des notions que nous aborderons dans la séance de TP 3.

Il est donc vivement conseillé de travailler entièrement cette fiche (**tout au moins le Thème 1**) avant la prochaine séance de TP, afin d'être à jour par rapport au cours.

Cette fiche est formée de trois parties :

- Thème 1 porte sur la méthode du point fixe et son application à la recherche de zéros de fonctions. L'objectif visé ici est la compréhension du cours.
- Thème 2 porte sur une application à un problème concret.
- Thème 3 porte sur des utilitaires. Ici des scripts sont fournis pour aider dans les mises en oeuvre.

Toutes les fonctions fournies dans les exercices sont identifiées à leur prolongement par continuité (aux points d'indétermination s'il y en a). Il en est de même de leurs dérivées.

Thème - 1 Autour de la méthode du point fixe

Thème - 1-1 La méthode et son analyse

Note 1 (Méthode du point fixe).

La méthode du point fixe construit une suite (x_n) approchant le point fixe l d'une fonction ϕ , par :

$$\begin{cases} u_{n+1} &= \phi(u_n), & n = 0, 1, \dots \\ u_0 & \text{donné} \end{cases} \quad (1)$$

Dans cette méthode, la fonction ϕ est appelée *fonction d'itération* et la valeur l telle que $l = \phi(l)$ est appelée *limite* de la suite (1) ou solution.

Dans les exercices qui suivent, nous allons observer le comportement de cette méthode suivant la valeur que prend la dérivée de la fonction d'itération au point fixe. *On rappelle qu'en pratique on ne dispose pas de ces informations. Mais celles-ci nous permettent de mieux comprendre le comportement de la méthode.*

Exercice-1 : Cas où $|\phi'(l)| > 1$

On considère la fonction $\phi(x) = x^2 - 1$ sur $[0, 2]$

Q-1 : Existence.

Représenter sur une même figure les courbes $x \mapsto \phi(x)$ et $x \mapsto x$. Représenter aussi le point de coordonnées $(\frac{1+\sqrt{5}}{2}, \frac{1+\sqrt{5}}{2})$. Conclure que la fonction ϕ admet un point fixe l dans $[0, 2]$.

Q-2 : Analyse.

Q-2-1 : Essayer d'appliquer la méthode du point fixe pour déterminer le point fixe l de ϕ . On prendra comme paramètre $u_0 = 1$, $\varepsilon = 10^{-3}$, $iterMax = 100$.

On pourra utiliser la fonction **pointFixe** voir Listing 2

Q-2-2 : La méthode converge-t-elle ? Vérifier avec plusieurs valeurs de $u_0 \neq \frac{1+\sqrt{5}}{2}$.

Q-2-3 : Calculer $\phi'(\frac{1+\sqrt{5}}{2})$ et justifier le comportement observé.

Exercice-2 : Cas où $0 < |\phi'(l)| < 1$ (Ce qui importe ici c'est le fait que $\phi'(l) \neq 0$).

On considère la fonction $\phi(x) = \frac{x^2 - x + x \log(x)}{x - 1 + x \log(x)}$ définie sur $[\frac{1}{2}, 2]$.

Q-1 : Existence.

En représentant sur une même figure les courbes $y = x$ et $y = f(x)$ déduire que cette fonction admet un point fixe $l = 1$.

Q-2 : Analyse.

Q-2-1 : En utilisant la méthode du point-fixe, déterminer ce point fixe et estimer la vitesse de convergence pour ce problème. On prendra comme paramètre $u_0 = 1.25$, $\varepsilon = 10^{-6}$, $iterMax = 100$.

(On pourra utiliser à cette fin la fonction **traquerOrdreSuite** voir Listing 3).

Q-2-2 : Représenter sur un même graphique la suite $\frac{|u_{n+1} - l|}{|u_n - l|}$ et la droite horizontale d'équation $y = |\phi'(l)|$. Justifier alors l'ordre de convergence observé ci-dessus.

Exercice-3 : Cas où $\phi'(l) = 0$ et $\phi''(l) \neq 0$

Dans cette partie on considérera l'une des fonctions suivantes :

- $\phi(x) = \frac{2x^3 + 2x^2 + 20}{3x^2 + 4x + 10}$, sur l'intervalle $[1, 2]$.
- $\phi(x) = \frac{x^2 - x + x(2-x)\log(x)}{x - 1 + x \log(x)}$, sur l'intervalle $[\frac{1}{2}, 2]$.

Q-1 : Existence.

Représenter sur le même graphique la courbe $x \mapsto \phi(x)$ et la courbe $x \mapsto x$, et conclure que la fonction ϕ admet un point fixe dans l'intervalle considéré.

Q-2 : Analyse.

Q-2-1 : En utilisant la fonction **pointFixe**, calculer la valeur approchée du point fixe l de ϕ , ainsi que la suite (u_n) des itérés générés par la méthode. On prendra comme paramètre $u_0 = 1.5$, $\varepsilon = 10^{-6}$, $iterMax = 100$.

Q-2-2 : La méthode du point fixe converge-t-elle pour ce problème ? Si oui donner l'ordre de convergence observé. (On pourra utiliser à cette fin la fonction **traquerOrdreSuite**.)

Q-2-3 : Calculer $\phi'(l)$ et $\phi''(l)$.

Q-2-4 : Représenter sur un même figure les suites $\frac{|u_{n+1} - l|}{|u_n - l|}$ et $\frac{|u_{n+1} - l|}{|u_n - l|^2}$.

Q-2-5 : Représenter sur la figure précédente les droites horizontales $y = |\phi'(l)|$ et $y = \frac{1}{2}|\phi''(l)|$.

Q-2-6 : Commenter le graphique obtenu. Les observations étaient-elles prévisibles ?

Note 2 (Méthode du point fixe et zéros de fonctions).

On souhaite appliquer la méthode du point fixe pour déterminer une racine l d'une fonction f (c'est-à-dire un point l tel que $f(l) = 0$).

Il existe des solutions à ce problème. On peut les classer selon l'ordre de dérivation de f qu'elles utilisent. Nous supposons dans ce qui suit, que nous disposons de l'expression de f' et souhaitons tirer partie de ce qui a été fait précédemment.

Nous optons alors pour la méthode du point fixe (1), dans laquelle la fonction d'itération est donnée par :

$$\phi(x) = x - \frac{f(x)}{f'(x)}. \quad (2)$$

On peut ainsi implémenter cette méthode sous Python en utilisant la méthode du point fixe :

```
def zeroParPointFixe(f, df, u0, epsi, iterMax, stockage):
    # df est la dérivée de f
    g = lambda x: (x - f(x)/df(x))
    return poinFixe(g, u0, epsi, iterMax, stockage)
```

Exercice-4 : **Prise en main**

Q-1 : **Identification de la méthode.**

Q-1-1 : En vous servant de votre cours identifier la méthode décrite dans la Note2 et appliquée à la détermination d'un zéro de f .

Q-1-2 : Exprimer $\phi'(l)$ et $\phi''(l)$ en fonction de $f(l), f'(l), f''(l)$. Conclure en vous inspirant de l'Exercice 3 ci-dessus que cette méthode convergera le plus souvent à d'ordre 2.

Q-1-3 : Aura-t-on encore l'ordre 2 si $f'(l) = 0$?

Q-2 : **Une implémentation.**

Implémenter cette méthode à travers une fonction Python **Newton** de prototype

Listing 1 – fichier Newton.py

```
def Newton(f, df, u0, epsi, iterMax, stockage)
```

où df désigne la dérivée f' de f et les autres paramètres sont comme dans la méthode **pointFixe** voir Listing 2. (On peut si on le veut utiliser la consigne de la Note2)

Exercice-5 : **Validations.**

Q-1 : **Validations I : un zéro de l'équation de Fibonacci voir TP2**

On pose $f(x) = x^3 + 2x^2 + 10x - 20$

Q-1-1 : Représenter sur $[1, 2]$ la courbe de f et en déduire que f admet un zéro l dans cet intervalle.

Q-1-2 : En utilisant la méthode de Newton décrite ci-dessus, déterminer l ainsi que la vitesse de convergence de cette méthode pour ce problème.

Q-1-3 : Calculer $\phi''(l)$ et justifier l'ordre de convergence observé.

Q-2 : Validations II : ordre lorsque $f'(l) = 0$.

On pose cette fois $f(x) = (x - 1)\log(x)$ sur $[\frac{1}{2}, 2]$.

Q-2-1 : Montrer que $l = 1$ est zéro double de f .

Q-2-2 : Appliquer la méthode de Newton décrite ci-dessus sur ce problème et vérifier qu'elle est convergente d'ordre 1 seulement !

Q-2-3 : Tentative de correction

Reprendre la question précédente en remplaçant la fonction d'itération par $\phi(x) = x - 2\frac{f(x)}{f'(x)}$.

Cela revient à remplacer sous Python l'appel

```
l0, n, u = Newton(f, df, u0, epsi, iterMax, stockage)
```

par le suivant

```
l0, n, u = Newton(lambda x: 2 * f(x), df, u0, epsi, iterMax, stockage)
```

Qu'observez-vous ?

Q-3 : Validations III : importance de la donnée initiale u_0 : convergence/divergence.

On pose cette fois $f(x) = \frac{x}{1+x^2}$ sur $[-1, 8]$

Q-3-1 : Représenter graphiquement la courbe de f sur cet intervalle.

Q-3-2 : Appliquer la méthode de Newton pour déterminer le zéro $l = 0$ avec les paramètres suivants : $u_0 = 2$, $\varepsilon = 10^{-6}$, $iterMax = 20$. Qu'observez-vous ?

Q-3-3 : Reprendre la question précédente avec $u_0 = \frac{1}{2}$, $\varepsilon = 10^{-6}$, $iterMax = 20$. Qu'observez-vous ?

Q-4 : Validations IV : importance de la donnée initiale u_0 : compréhension de la méthode.

On considère l'équation $f(x) = 0$, avec $f(x) = x(x^2 - 1)$.

Q-4-1 : Représenter graphiquement la courbe de la fonction f sur $[-2, 2]$.

Q-4-2 : On souhaite utiliser la méthode de Newton pour déterminer la racine $x = 1$ de cette équation. Expliquer comment choisir $u_0 \neq 1$ pour que cette méthode converge en une seule itération. Vérifier le résultat numériquement.

Q-4-3 : De la question précédente, expliquer comment choisir $u_0 \neq 1$ pour que cette méthode converge en exactement deux itérations. Vérifier le résultat numériquement.

Thème - 2 Application à un problème concret : Estimation de la vitesse limite de chute d'une balle de base-ball

Note 3 (Attention !!!).

Le présent exercice montre un exemple d'application de la méthode de Newton. Il n'est pas obligatoire pour la séance de TP.

On considère une sphère de masse m et de diamètre d en chute dans un champ de pesanteur de constante g en présence des frottements non négligeables de l'air. On s'intéresse alors à sa vitesse limite v supposée positive.

En appliquant le principe fondamental de la dynamique du point, on montre que la vitesse limite de la sphère est donnée par l'équation :

$$v^2 = \frac{2mg}{\rho S c_D} \quad (3)$$

où ρ est la densité de l'air, $S = \pi d^2/4$ la surface frontale de la sphère, c_D le *coefficient de traînée*, obtenu expérimentalement et qui pour une sphère a l'expression :

$$c_D = \frac{24}{Re} + \frac{6}{1 + \sqrt{Re}} + \frac{2}{5} \quad (4)$$

où Re appelé *nombre de Reynolds* s'exprime en terme de la vitesse par la formule :

$$Re = \rho v d / \mu \quad (5)$$

μ étant la *viscosité dynamique* de l'air.

En combinant ces expressions on obtient une équation sur Re dont la résolution permet de déterminer la vitesse v .

Dans la suite on prendra pour les différentes constantes les valeurs suivantes (données dans le Système International d'unités) :

- pour l'air $\mu = 1.8 \times 10^{-5}$, $\rho = 1.2$, $g = 9.8$
- pour un ballon de base-ball : $d = 0.075$, $m = 0.14$

Exercice-1 : Existence de solution

Q-1 : Montrer que (3) peut se mettre sous la forme : $(Re)^2 c_D = \alpha$ où α est indépendante de v ou Re .

Q-2 : En déduire l'équation $f(Re) = 0$ à résoudre pour déterminer Re . Expliquer comment déterminer v une fois cette équation résolue.

Q-3 : En remarquant que l'équation ci-dessus aurait pu aussi s'écrire $c_D Re = \frac{\alpha}{Re}$, représenter graphiquement sur une même figure les courbes associées à chaque membre de cette dernière équation et en déduire que le problème considéré admet une solution Re située dans l'intervalle $]0, \frac{\alpha}{24}[$.

Exercice-2 : Résolution numérique

On souhaite résoudre l'équation $f(Re) = 0$ par la méthode de Newton.

Q-1 : A la lueur des graphiques de l'exercice précédent, expliquer comment choisir le point de départ.

Q-2 : Résoudre l'équation en utilisant la méthode de Newton. Puis déterminer la vitesse limite v .

Q-3 : Comparer la vitesse obtenue à celle du record dans le circuit professionnel de base-ball. (Recherche Google avec mot clé **fastball**)

Thème - 3 Quelques Scripts

Listing 2 – fichier pointFixe.py

```
def pointFixe(g, u0, epsi, iterMax, stockage):
    """
    ENTREES :
        g      : fonction dont on cherche le point fixe
        u0     : point de depart de la methode du point fixe
        epsi   : tolerance. On n'arrete les calculs si |u_n+1 - u_n| < epsi
        iterMax : nombre maximum d'iterations autorisees
        Stocake : si different de 0, on stocke tous les u_n
    SORTIES :
        u_iter : la limite
        n_iter  : le nombre d'iterations effectuees
        u       : la liste des u_n si stockage different de 0
    """
    # Initialisation
    u_iter = u0 # iterateur: donnee
    num_iter = 0 # iterateur: indice
    u = [u_iter] # stockage
    conv = False # test d'arret
    # Iterations
    while (not(conv)):
        # Chargement des donnees
        u_in = u_iter
        # Calcul
```

```

    u_out = g(u_in)
    # Mise a jour
    num_iter = num_iter + 1
    u_iter = u_out
    # Stockage
    if(stockage != 0):
        u.append(u_iter)
    # Post-traitement
    err = abs(u_iter - u_in)
    print('iterations {0:d} erreur {1:e}'.format(num_iter, err))
    # Test de convergence
    conv = ((err < epsi) or (num_iter > iterMax))
    return u_iter, num_iter, u

def test_pointFixe():
    g = lambda x: (x+1)/2
    u0, epsi, iterMax, stockage = 0, 1e-5, 20, 1
    lim, n, u = point_fixe(g, u0, epsi, iterMax, stockage)
    print('meth = Point fixe: lim = {0:10.9e}: u0 = {1:f} : prec={2:5.4e} :\
iter= {3:3d}'.format(lim, u0, epsi, n))

if __name__ == '__main__':
    test_pointFixe()

```

Listing 3 – fichier traquerOrdreSuite.py

```

# -*- coding: utf-8 -*-
"""
Created on Wed Mar  1 13:56:12 2017

@author: apoung
"""
import numpy as np
import matplotlib.pyplot as plt

def traquerOrdreSuite(suite, lim, numFig=1):
    """
    suite : suite generee : doit etre un numpy array
    lim : limite de la suite ou solution exacte
    numFig : numero de la figure de sortie
    Modifier cette fonction si l'ordre de convergence peut etre superieur a 3
    """
    # err = np.abs(np.array(suite,dtype=float128) - lim)
    epsi = 1e-11
    err = np.abs(np.array(suite) - lim)
    n = len(err)
    plt.figure(numFig)
    plt.clf()
    x = err[0:n-1]
    y = err[1:n]
    # Affichage graphique de l'ordre (Recommande !!!)
    plt.loglog(x, y, '-+b', x, x, 'r', x, x**2, 'g', x, x**3, 'y')
    plt.legend(['nuage', 'pente=1', 'pente=2', 'pente=3'], loc='bottom')
    # Affichage sous forme de text de l'ordre (Pas obligatoire !!!!!!!)
    calculPossible = False
    while(not(calculPossible) and (n > 0)):
        calculPossible = np.abs(np.log(x[n-2]) - np.log(x[0])) > epsi
        n = n - 1
    if(calculPossible):
        ordre = (np.log(y[n-2]) - np.log(y[0]))/(np.log(x[n-2]) - np.log(x[0]))
        titre = "Convergence d'ordre: {0:3.1f}".format(ordre)
        print(titre)
        plt.title(str(titre))
    else:
        plt.title("Ordre de convergence")

def test_traquerOrdreSuite():
    n = np.arange(7)
    x = (0.9)**(2**n) # suite d'ordre 2
    y = (0.9)**(3**n) # suite d'ordre 3
    traquerOrdreSuite(x, 0, 1)
    traquerOrdreSuite(y, 0, 2)

if __name__ == '__main__':
    test_traquerOrdreSuite()

```