Département de Mathématiques 2018/2019
Module: Analyse numérique 1
Fiche de T.P. N°4

Résolution numérique des équations non linéaires

Méthode de bissection

Supposons que vous souhaitiez trouver une solution à $f(x) = 0$ à l'aide de la méthode de bissection et que vous disposiez d'une fonction **Python** qui évalue $f(x)$ pour un x donné. Supposons également qu'on vous ait fourni un intervalle $[a, b]$ et une tolérance tol. Voici le programme de base pour la bissection:

```
def bissect(f,a,b,tol=1e-6):
    c = (a + b)/2
    max_err = (b-a)/2
    while max_err > tol:
        if f(c) == 0:
            break # sortie de la boucle while
        elif f(a)*f(c) < 0:
            b = c
        else:
            a = c
    # nouvelle estimation
    c = (a + b)/2
    max_err = (b-a)/2
    return c
```

1. Ecrire la fonction bissect en demandant au programme d'afficher le nombre d'itérations pour arriver au résultat avec la précision voulue.
2. Faire le graphique des fonctions suivantes de sorte à s'assurer qu'elles n'ont qu'un seul zéro. Calculer ensuite ce zéro par la méthode de la bissection.
   i) $f(x) = x^5 - 2x^4 + 100x^3 - 2$
   ii) $f(x) = x + e^x$
   iii) $f(x) = x - 1 + \frac{1}{2}\sin x$
3. Si la fonction f est continue et $f(a)f(b) < 0$ la méthode de bissection converge cependant cette convergence peut être très lente. Pour éviter ce problème, ajouter une condition supplémentaire à la déclaration while: si le nombre d'itération est supérieur à une limite max_iter quitter la boucle et la fonction peut renvoyer **None (**pour abscence de valeur**).**

Méthode du point fixe

1- Trouver un critère d'arrêt pour la méthode du point fixe appliquer à l'équation

$$g(x) = x.$$

2- Ecrire une fonction Python, function **pointfixe(g,x0,tol, nmax)** calculant la solution de $g(x) = x$ par la méthode du point fixe.
3- Calculer la première solution positive de $\cos(x) = x$.
4- On souhaite maintenant calculer la première solution positive de $\tan(x) = 1/x$ par la méthode du point fixe. Pour cela on pose:

$$g(x) = x - \alpha(x * \tan(x) - 1), \qquad \text{où } \alpha > 0$$

Essayer la méthode du point fixe pour $\alpha \in \{0.1, 1, 0.01\}$. Expliquer les résultats.
5- Tracer l'erreur $\log_{10}(|f(x_n)|)$ en fonction de n, où f est la fonction $f(x) = x\tan(x) - 1$
6- Comparer les vitesses de convergence des différents algorithmes.

Méthode de Newton

1- Ecrire une fonction Python **newton(f,df,x0,eps,nmax)** calculant la solution de $f(x) = 0$ par la méthode de Newton.
2- Utiliser la méthode de Newton pour calculer le zéro de $f(x) = \frac{1}{x} + \ln(x) - 2, \qquad x > 0$
Comparer à la valeur obtenue avec la fonction **newton** de **scipy.optimize.**
3- Modifier la fonction newton de sorte à pouvoir représenté graphiquement l'erreur $\log_{10}(|f(x_n)|)$ en fonction des itérations n.
Tracer l'erreur $\log_{10}(|f(x_n)|)$ en fonction de n sur la figure(3).
4- Utiliser la méthode de Newton pour calculer le zéro de $f(x) = (x - 1) * \ln(x)$ sur [0,1]. Tracer l'erreur $\log_{10}(|f(x_n)|)$ en fonction de n sur la même figure(3).
Comparer avec les courbes obtenues et expliquer pourquoi la convergence n'est pas quadratique dans le second cas.