TP Informatique n° 6
Révisions : méthode de Newton et simulations

1 Résolution d'équation à l'aide de la méthode de Newton

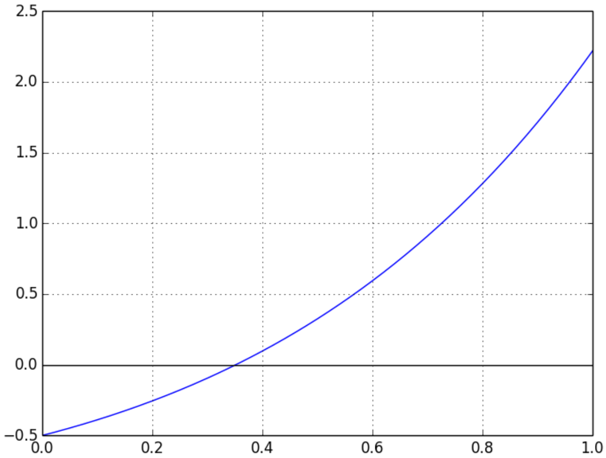
Soit f une fonction de classe $\mathcal{C}^2$ sur un intervalle ouvert I, c un point de I tel que $f(c) = 0$ et $f'(c) \neq 0$. On admet la proposition suivante :

Proposition 1 Il existe un intervalle ouvert J contenant c tel que pour tout $x \in J$, la suite récurrente (x_n) définie par :

$$\begin{cases} x_0 &= x \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \end{cases}$$

converge vers c.

On ne se préoccupera pas de la justification de la convergence de cette méthode mais on l'observera numériquement sur des exemples.



■ **Exemple 1** Ecrire l'équation de la tangente à la courbe $\mathcal{C}_f$ en un point d'abscisse $x_0 = 0.8$ et déterminer son intersection avec l'axe des abscisses. En déduire graphiquement le comportement des premiers termes de la suite. ■

■ **Exercice 1** Ecrire une fonction `newton(f,df,x0,eps)` prenant en paramètre une fonction `f`, sa dérivée `df`, une valeur initiale x_0 et un réel $\varepsilon > 0$, et qui renvoie une valeur approchée de la solution de l'équation $f(x) = 0$. On prendra comme condition d'arrêt : $|x_{n+1} - x_n| \leqslant \varepsilon$. ■

■ **Exercice 2** On souhaite programmer la méthode de Newton sans connaitre l'expression de la dérivée.

1. Montrer à l'aide de la formule de Taylor Young que si f est de classe $\mathcal{C}^3$, quand $h \to 0$:

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(x)}{2}h + o(h) \text{ et } f'(x) = \frac{f(x+h) - f(x-h)}{2h} = f'(x) + o(h)$$

2. Ecrire une fonction `newtonbis(f,x0,h,eps)` n'utilisant pas l'expression de f' pour le calcul.

■

Remarque 1 Comparer avec la fonction newton de la bibliothèque scipy.optimize.

2 Exercices de simulation

■ **Exercice 3** Soit n un entier naturel non nul. Une urne contient $2n$ boules indiscernables au toucher :
- n sont numérotées 0
- les autres sont numérotées de 1 à n

On effectue au hasard deux tirages successifs et sans remise d'une boule dans cette urne. On note X le plus grand numéro obtenu et Y le plus petit.

1. Déterminer les valeurs prises par X.
2. Proposer une représentation informatique de l'urne et écrire une fonction en Python permettant de simuler l'expérience aléatoire décrite, et renvoyant la liste `[X,Y]`. On testera la fonction pour plusieurs valeurs de n.
3. Proposer une fonction permettant de simuler m expériences et retournant la liste des résultats de ces m expériences. Tester la fonction pour différentes valeurs de m et de n.
4. Ecrire une fonction permettant d'évaluer la probabilité $\mathbf{P}(X = Y + 1)$ à l'aide d'une fréquence. Que vaut la probabilité (théorique) $\mathbf{P}(X = Y + 1)$?
5. Proposer une illustration graphique de la question précédente.

■

Exercice 4 On considère une urne contenant une boule rouge et une boule verte. On effectue N tirages successifs : on tire une boule de l'urne, que l'on remet avec une boule de la même couleur. On note X_N le nombre de boules rouges après ces N tirages.

1. Ecrire en Python une fonction`X(N)` qui renvoie la valeur de X_N.
2. Prendre $N = 10$ et effectuer 100 simulations de X_N. Afficher alors la moyenne des valeurs obtenues. Exécuter plusieurs fois ces instructions et comparer ces moyennes empiriques. Que devient cette moyenne lorsqu'on effectue 10000 simulations de X_N ?
3. Conjecturer la loi de X_N. On pourra afficher les fréquences d'apparition en faisant un diagramme en bâtons.

■