Informatique Appliquée au Calcul Scientifique 1

Séance 9

Algorithme de Newton

Table des matières

I. Le principe de l'algorithme de Newton

L'algorithme de Newton, également appelé méthode de Newton-Raphson, est une technique itérative utilisée pour trouver les racines d'une fonction réelle $f(x)$. Cette méthode est particulièrement efficace pour des fonctions dérivables et convergent rapidement, surtout lorsque l'estimation initiale est proche de la racine.
À chaque itération, la fonction pour laquelle on recherche une racine est approximée linéairement autour de l'itération courante (ou point courant), et l'itération suivante est définie comme étant le zéro de cette approximation linéaire. Cette description simplifiée suggère que deux conditions sont essentielles pour le bon fonctionnement de l'algorithme : la fonction doit être dérivable aux points visités (pour permettre cette approximation linéaire) et les dérivées en ces points ne doivent pas être nulles (afin que la fonction linéarisée possède un zéro).
On va donc construire une bonne approximation d'une racine de la fonction f en considérant son développement au premier ordre.

Formule de Taylor :
Soient I un intervalle réel, a un élément de I et f une fonction dérivable en a jusqu'à un certain ordre $n \geq 1$.
Alors, pour tout nombre réel x appartement à I, on a :

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f^{(2)}(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_n(x)$$

où $R_n(x)$ est une fonction négligeable par rapport à $(x-a)^n$ au voisinage de a.

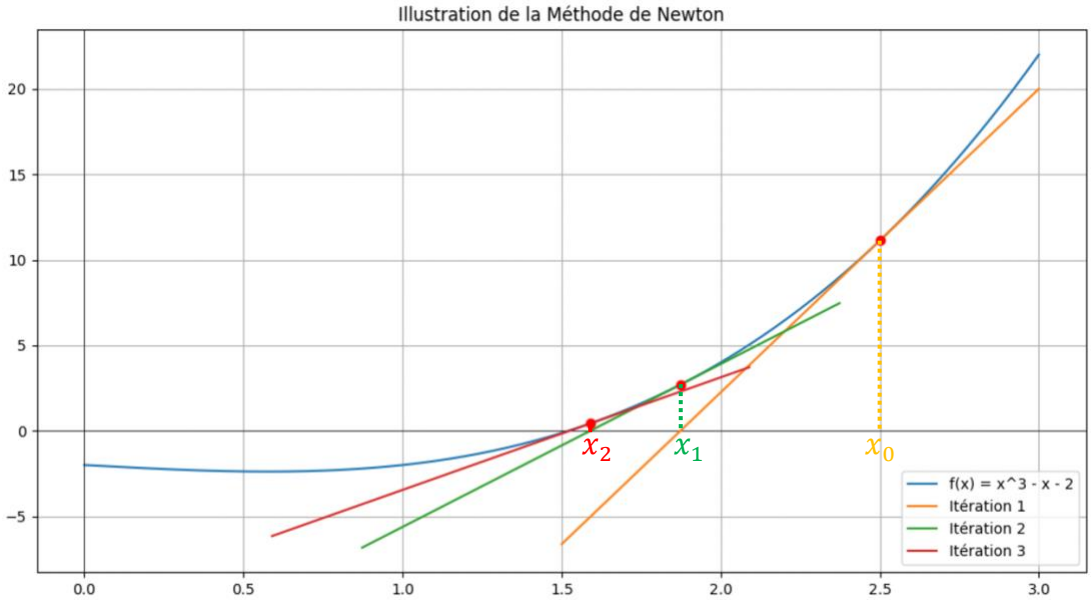
Ainsi, à l'ordre 1, on considère la fonction environ égale à sa tangente en ce point :

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

x_0 étant notre premier point que l'on choisira proche de la racine recherchée.
Ensuite, nous allons chercher le zéro de notre approximation, ce qui se fera simplement en résolvant :

$$f(x_0) + f'(x_0)(x - x_0) = 0$$

Ce qui nous donnera ainsi x_1 plus proche du zéro recherché ($x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$).
Puis nous approcherons de nous la fonction par sa tangente en x_1. Et ainsi de suite.



La méthode :
Après avoir choisi x_0 proche du zéro de la fonction étudié, on construit par récurrence la suite :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Remarque :
Avec cette méthode, la dérivée doit pouvoir être calculer et ne pas s'annuler.

II. Exemples d'application

II.1. Résolution de $x^3 + x - 1 = 0$

C'est une fonction strictement croissante sur $\mathbb{R}$, et elle admet une racine réelle sur $[0\ ;\ 1]$.
Sa dérivée est : $f'(x) = 3x^2 + 1$ et elle ne s'annule jamais.
On a alors :

$$x_{n+1} = x_n - \frac{{x_n}^3 + x_n - 1}{3{x_n}^2 + 1}, n \geq 0$$

Cette suite converge rapidement vers la racine recherchée.

II.2. Calcul de racine de *a*

On peut calculer avec l'algorithme de Newton le réel $\sqrt{a}$, qui vérifie l'équation $x^2 - a = 0$, mais aussi $x - \frac{a}{x} = 0\ , x > 0$.
En posant $f(x) = x - \frac{a}{x}$, on a $f'(x) = 1 + \frac{a}{x^2}$. $f'(x)$ n'est jamais nul pour $x > 0$.
On pourra travailler avec la relation :

$$x_{n+1} = x_n - \frac{x - \frac{a}{x_n}}{1 + \frac{a}{{x_n}^2}}, n \geq 0\ et\ x_0 > 0$$

II.3. Calcul de π

On peut approcher π grâce à la relation : $\tan\left(\frac{\pi}{4}\right) = 1$.
Soit, $f(x) = \tan(x) - 1\ ,\ -\frac{\pi}{2} < x < \frac{\pi}{2}$.
On a alors, $f'(x) = \frac{1}{cos^2(x)}$ et pour utiliser l'algorithme de Newton, on posera :

$$x_{n+1} = x_n - cos^2(x)(\tan(x) - 1), x_0 = 1$$

Il ne faudra pas oublier de multiplier le résultat final par 4…

III. TP4 – Algorithme de Newton

1. Mettre en place sous Python, une fonction `Newton` qui prendra comme arguments la fonction, sa dérivée le nombre d'itérations voulu et le nombre de départ.

2. Tester votre fonction pour la résolution de $x^3 + x - 1 = 0$, et trouver une valeur pour $\sqrt{2}$ et π.
3. Calculer numériquement toutes les racines de $1 + 3x - x^3 = 0$.
4. Bonus : optimiser votre algo avec un argument supplémentaire qui sera l'écart entre 2 éléments de la suite et lever une erreur si la dérivée devient négative.