# DST 3 IPT

**Exercice 1 - Méthode de Newton.** On se donne une fonction $f : \mathbb{R} \to \mathbb{R}$, dont on cherche à calculer un point d'annulation c'est-à-dire une solution de l'équation $f(x) = 0$.

1. Vous avez totalement oublié la commande pour calculer la valeur absolue. Il faut la reprogrammer : proposer une fonction $val\_abs(x)$ qui renvoie $x$ lorsque $x \geqslant 0$ et $-x$ sinon.
2. (Pas de code). L'alglorithme de Newton pour approcher des solutions de l'équation $f(x) = 0$ est le suivant : supposons qu'on soit arrivé à une valeur $u_n \in \mathbb{R}$, on construit $u_{n+1}$ comme suit :
   - On considère la tangente à la courbe de $f$ au point d'abscisse $u_n$.
   - On note $u_{n+1}$ l'abscisse de l'intersection de cette tangente avec l'axe $(Ox)$.

   Illustrer, et montrer que $u_{n+1}$ vérifie la relation
   $$u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}.$$
3. On suppose que la fonction $f$ et sa dérivée $f'$ ont été codées, et sont notées en python $f$ et $fp$. Proposer une fonction $Newton(a, E)$ qui calcule les valeurs de la suite définie par
   $$u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)} \quad \text{et} \quad u_0 = a,$$
   jusqu'à ce que $|u_n| < E$. On ajoutera un critère d'arrêt au cas où ce critère n'est pas atteint : on limitera l'algorithme à 100 itérations au maximum. La fonction renverra la dernière valeur calculée.
4. On peut montrer que l'algorithme converge très vite, sous certaines conditions, en particulier, à condition de partir d'une valeur $a$ déjà assez proche d'une solution de l'éqution $f(x) = 0$. Quel autre algorithme connaissez-vous pour approcher numériquement les solutions de cette équation ? (On attend juste le nom d'un tel algorithme).

**Exercice 2 - Un tri naïf.** Etant donné une liste $L$, voici un algorithme de tri très naïf :

- On parcourt la liste du début à la fin, si deux éléments consécutifs, aux places $j$ et $j+1$, vérifient $L[j+1] < L[j]$, on les permute. A la fin de ce parcours, le plus grand élément de $L$ est « remonté » (tel une bulle) à la dernière place.
- On réapplique ce procédé jusqu'à ce que la liste soit triée (noter qu'à chaque étape on n'a pas besoin de l'appliquer jusqu'au bout, car la fin de la liste est déjà triée).

On rappelle qu'on peut affecter simultanément à deux variables $u$ et $v$ les valeurs $a$ et $b$ avec la ligne de code $u, v = a, b$

Ecrire une fonction $Tri\_bulle$ qui prend en argment une liste et la trie par l'algorithme décrit ci-dessus. Votre fonction indiquera en outre le nombre de permutations ayant eu lieu dans la liste.