TP Informatique commune 1A : Calcul approché de racine p-ième

Soit A un réel strictement supérieur à 1 et p un entier naturel supérieur ou égal à 2. On pose $f_p(x) = x^p - A$. On remarque que l'équation $f_p(x) = 0$ admet une unique solution $\alpha = A^{1/p}$ sur $[1, +\infty[$. Le but du TP est d'obtenir une valeur approchée de α à l'aide de deux méthodes : la dichotomie et la méthode de Newton.

Exercice 1 *Méthode de dichotomie*

On se place sur un intervalle $[a_0, b_0]$ tel que $f_p(a_0)$ et $f_p(b_0)$ soient de signes opposés (ainsi $\alpha \in [a_0, b_0]$ par continuité de f_p). On construit alors deux suites récurrentes (a_n) et (b_n) de la manière suivante:

- On pose $c = (a_n + b_n)/2$ le milieu de $[a_n, b_n]$.
- Si $f_p(c)$ est du même signe que $f_p(a_n)$ on pose $a_{n+1} = c$ et $b_{n+1} = b_n$.
- Sinon on pose $b_{n+1} = c$ et $a_{n+1} = a_n$.

On obtient ainsi un nouvel encadrement de α dont la taille est la moitié du précédent. On itère ce procédé le nombre de fois souhaité ou jusqu'à obtenir une précision donnée. Ici, nous allons nous arrêter au bout d'un certain nombre d'étapes N donné en entrée (voir le cours pour la version avec la précision donnée). Rappelons qu'au bout de n étapes la précision obtenue est $\frac{b_0 - a_0}{2^n}$.

1. Écrire une fonction `dicho(p,A,N)` qui prend en entrée la puissance p, le réel A dont on cherche la racine p-ième et un nombre N d'étapes, et qui retourne la liste `L` des encadrements de α obtenus par dichotomie à partir du segment $[1, A]$. `L[0]` sera la liste des bornes inférieures a_n et `L[1]` sera la liste des bornes supérieures b_n.

2. Écrire un programme demandant à l'utilisateur les entiers p et N et le réel A et qui affiche le dernier encadrement de α obtenu ainsi que le graphique correspondant : on tracera la courbe sur f_p, et on représentera par des ronds sur la courbe les bornes inférieures, et par des croix les bornes supérieures.

   Compléter le tableau avec les derniers encadrements obtenus:

p	A	N	borne inf	borne sup
3	156	18		
20	543576	39		

Exercice 2 *Méthode de Newton*

On construit une suite (x_n) de valeurs approchées de la façon suivante:

- On choisit $x_0 \in [1, +\infty[$.
- On pose x_1 le point d'intersection de l'axe des abscisses avec la tangente à la courbe de f au point d'abscisse x_0.
- On construit de même les termes suivants de la suite.

On rappelle que l'on a alors la relation de récurrence : $x_{n+1} = x_n - \frac{f_p(x_n)}{f_p'(x_n)}$.

1. Écrire une fonction `newton(p,A,x0,N)` qui retourne la liste `L` des coordonnées des points construits par une méthode de Newton en N étapes en partant de x_0. Remarque: pour calculer la dérivée, ne pas utiliser une méthode numérique intégrée à Python mais entrer directement la formule de f_p'.

2. Écrire un programme demandant à l'utilisateur les valeurs de p, A, x_0 et N et qui affiche la dernière valeur approchée de α obtenue ainsi que le graphique correspondant : courbe la fonction f_p ainsi que les points sur la courbe de la méthode de Newton.

   Compléter le tableau avec les dernières valeurs approchées obtenues. Commenter.

p	A	x_0	N	valeur approchée
3	156	1	10	
3	156	156	10	
3	156	156	11	
20	543576	543576	100	
20	543576	1	100	
20	543576	543576	200	
20	543576	1	200	

Exercice 3 *Comparaison des deux méthodes*

1. Écrire une fonction `compareNewtonDicho(p,A,N)` qui retourne le nombre d'étapes nécessaires par dichotomie pour obtenir une valeur approchée supérieure de α au moins aussi fine qu'avec une méthode de Newton en N étapes initialisée avec $x_0 = 1$.

2. Compléter les tableaux suivants:

   - Pour $p = 2$ et $A = 15$:

N	3	4	5	6
`compareNewtonDicho(p,A,N)`				

   - Pour $p = 5$ et $A = 7384$:

N	27	28	29	30
`compareNewtonDicho(p,A,N)`				

Exercice 4 *Fusion des deux méthodes*

On constate que si la méthode de Newton peut être plus lente sur les premières itérations, elle semble être beaucoup plus rapide que la dichotomie à partir d'un certain rang. Pour optimiser notre recherche de racine, on va donc fusionner les deux méthodes, c'est-à-dire appliquer à chaque itération les deux méthodes et conserver la valeur approchée supérieure la plus fine. Pour cela, on construit trois suites (a_n), (b_n) et (x_n) de la façon suivante:

Initialisation:

On pose $a_0 = 1$, $b_0 = A$, $x_0 = 1$.

Hérédite:

Pour tout $n \in \mathbb{N}$:

- On pose $c = \frac{a_n + b_n}{2}$ et $x_{n+1} = x_n - \frac{f_p(x_n)}{f_p'(x_n)}$.

- Si $f_p(c)$ est du même signe que $f_p(a_n)$ on pose $a_{n+1} = c$ et $b_{n+1} = b_n$, sinon on pose $b_{n+1} = c$ et $a_{n+1} = a_n$.
- Si $x_{n+1} < b_{n+1}$ on remplace b_{n+1} par x_{n+1}, sinon on remplace x_{n+1} par b_{n+1}.

1. Écrire la fonction `fusion(p,A,N)` qui retourne une valeur approchée de α en N étapes en fusionnant la recherche par dichotomie et la méthode de Newton (on retournera la valeur finale de x_N).

2. Écrire la fonction `compareFusionDicho(p,A,N)` qui retourne le nombre d'étapes nécessaires avec la méthode de dichotomie pour obtenir une valeur approchée supérieure de α au moins aussi fine qu'avec la méthode de fusion en N étapes.

3. Écrire la fonction `compareFusionNewton(p,A,N)` retourne le nombre d'étapes nécessaires avec la méthode de Newton (initialisée à $x_0 = 1$ et en ayant déjà fait une première étape) pour obtenir une valeur approchée supérieure de α au moins aussi fine qu'avec la méthode de fusion en N étapes.

4. Compléter le tableau suivant pour $p = 7$ et $A = 10^{10} + 1$:

N	10	20	25	27	30	31	32
`compareFusionDicho(p,A,N)`							
`compareFusionNewton(p,A,N)`							

Exercice 5 *Vitesse de convergence de l'approximation par dichotomie*

On cherche dans cette section à établir le nombre d'étapes à effectuer pour obtenir une valeur approchée de α avec une erreur inférieure à une quantité donnée.

On remarque qu'en l'initialisant sur $[1, A]$, on obtient au bout de n étapes une précision de $\frac{A-1}{2^n}$, qui est en particulier **indépendante de** p.

1. Écrire une fonction `vitesseDicho(A,n)` qui retourne le nombre d'étapes nécessaires pour obtenir par dichotomie à partir du segment $[1, A]$ une valeur approchée de α avec une erreur inférieure à 10^{-n}.

2. Compléter:

n	0	2	4	6	8	10
`vitesseDicho(4000,n)`						

Exercice 6 *Vitesse de convergence de l'approximation par la méthode de Newton*

On suppose ici $p = 2$. Soit A un réel strictement supérieur à 1 et (x_n) la suite d'approximation de $\alpha = \sqrt{A}$ obtenue par la méthode de Newton initialisée à $x_0 = 1$. On peut montrer les deux propriétés suivantes:

- Pour tout $n \in \mathbb{N}$, $|x_n - \alpha| \leqslant \frac{A-1}{2^n}$ (on converge au moins aussi vite que la dichotomie)
- Pour tout $n \in \mathbb{N}$, $|x_{n+1} - x_n| \leqslant \frac{1}{2}|x_n - \alpha|^2$ (convergence quadratique)

1. À l'aide des inégalités précédentes, écrire une fonction `vitesseNewton(A,n)` qui retourne le nombre d'étapes nécessaires pour obtenir par la méthode de Newton une valeur approchée de α avec une erreur inférieure à 10^{-n}.

2. Compléter:

n	0	2	4	6	8	10
`vitesseNewton(4000,n)`						