

DS3 – Partie écrite

Durée : 1 heure

Rappel des consignes

Lorsqu'on écrit un code Python, :

- faire attention à ce que les indentations soient visibles sur la copie ;
- commenter le code de façon à expliquer les grandes étapes de l'algorithme en ajoutant un commentaire en fin de ligne de code après le symbole `#`.

I Étude pas à pas d'un algorithme donné

```
1 def algorithme(f,a,b,eps):  
2     while (b-a)>2*eps:  
3         c=(float(a)+b)/2  
4         if f(a)*f(c)<0:  
5             b=c  
6         else:  
7             a=c  
8     return((a+b)/2)  
9  
10 def carre(x):  
11     return(x**2-3)
```

Programme 1: Programme à étudier.

1. Quel est le nom de l'algorithme réalisé par la fonction `algorithme` ?
2. Écrire une ligne de code python supplémentaire permettant au programme d'afficher à l'écran la valeur de la racine carrée de 3 à 0,15 près en partant de $a = 0$ et $b = 2$.
3. On exécute le programme contenant la ligne de code précédente. Détailler, pas à pas pour chacune des itérations de la boucle `while`, les valeurs de a et b en début d'itération et en fin d'itération. On présentera le raisonnement complet permettant de comprendre et de justifier l'évolution des valeurs à l'intérieur d'une itération.
4. Quelle est la valeur affichée à l'écran par le programme ?
5. Que se passe-t-il si on part de $a = 2$ et $b = 3$? Justifier.
6. Modifier le programme précédent pour que la fonction `algorithme` affiche un message d'erreur dans ces cas là. On donnera les lignes de code python utiles et on indiquera entre quelles lignes du programme proposé il faudrait les insérer.

Données : $1,25^2 = 1,5625$; $1,5^2 = 2,25$; $1,75^2 = 3,0625$; $2,25^2 = 5,0625$; $2,5^2 = 6,25$; $2,75^2 = 7,5625$.

II Question de cours – Méthode de Newton

On veut écrire un algorithme utilisant la méthode de Newton avec comme critère d'arrêt un nombre donné d'itérations.

Écrire en python une fonction `Newton` qui prend comme entrée la fonction à étudier, sa dérivée, une valeur initiale de la suite x_0 et un nombre d'itérations n et qui renvoie x_n approximation de la solution.

III Exercice – Méthode de la fausse position

Pour résoudre une équation non linéaire, la méthode de la fausse position s'inspire de la méthode de la dichotomie, en cherchant à mieux choisir le point de découpe de l'intervalle. Plutôt que de couper au milieu, la méthode de la fausse position choisit de couper à l'intersection entre l'axe des abscisses et la droite reliant les deux points de la courbe aux extrémités de l'intervalle (voir figure 1).

Par ailleurs, si la convexité de la courbe ne change pas, il est facile de voir que seule une borne de l'intervalle se déplace, l'autre restant fixe. Le critère d'arrêt n'est donc pas fixé sur la taille de l'intervalle mais est défini à partir de la distance entre deux points de découpe successifs, qui doit être inférieure à la précision demandée.

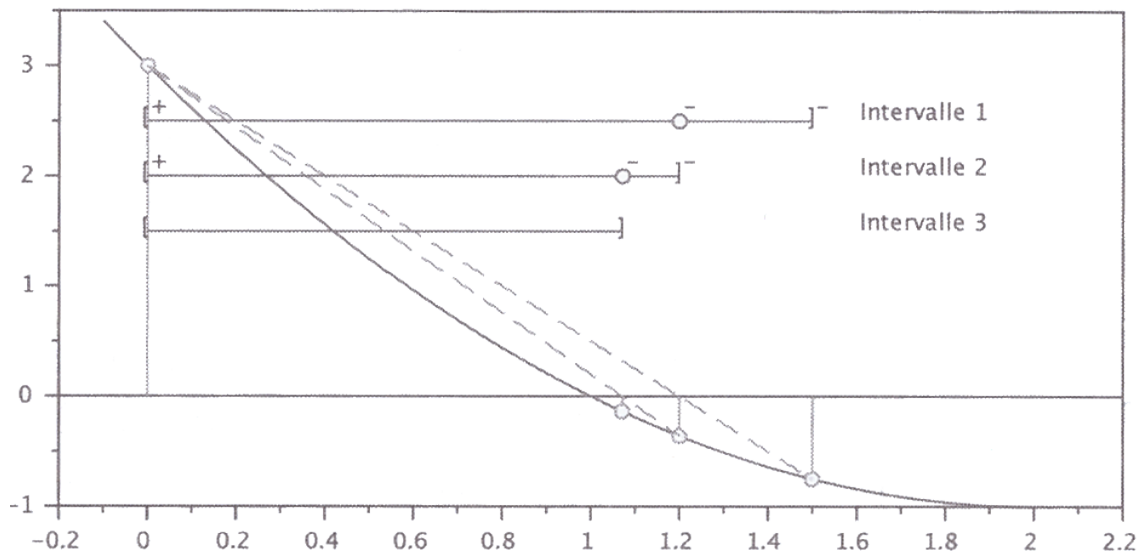


FIGURE 1 – Illustration de la méthode de la fausse position.

1. En considérant l'intervalle $[a; b]$ et une fonction f , déterminer l'expression de l'abscisse du point d'intersection entre l'axe des abscisses et la droite qui s'appuie sur les points de la courbe étudiée d'abscisses a et b .
2. On se place dans le cas d'une fonction quelconque, qui n'est pas nécessairement convexe. Écrire en python le code d'une fonction permettant de mettre en œuvre la méthode de la fausse position grâce à un algorithme dérivé de la méthode de dichotomie.