Exercices d'application

Problèmes stationnaires

Sources :

Savoirs et compétences :

- ❒ *SN.S2 : Problème stationnaire à une dimension. Méthode de dichotomie, méthode de Newton.*

Comparaison des méthodes de résolution

Soit l'équation E : $x\ln(x)=1$.

On admet (si un doute vérifiez-le rapidement) que E admet une unique solution α sur $\mathbb{R}$ et que $\alpha\in[1;e]$. On prendra e≈ 2.8. On peut aussi noter qu'en python `e` est un attribut de la bibliothèque `math`. On y a donc accès par `math.e`.

Méthode de dichotomie

Question 1 *Vérifier qu'à 10^{-6} près : $\alpha\approx 1.763223$.*

Question 2 *Modifier cette fonction afin qu'elle affiche le nombre de répétition de la boucle.*

Méthode des cordes (Lagrange)

Question 3 *À partir de la fonction précédente, créer une nouvelle fonction nommée `lagrange` renvoyant une valeur approchée à `epsilon` près de cette solution par la méthode des cordes (dite de Lagrange).*

Question 4 *Affichez le nombre de répétition de la boucle. Comparer avec la méthode précédente.*

Méthode de Newton

On souhaite comparer avec la méthode Newton, mais le test d'arrêt ne permet pas d'obtenir une valeur approchée avec un précision donnée. Nous allons donc nous servir des résultats précédents.

`f` et `fp` étant des fonctions saisies en python, et correspondant à f et f' alors l'algorithme de Newton est donné ci-contre.

Question 5 *Implémenter la méthode de Newton dans une fonction nommée `newton`.*

Question 6 *Comparer les méthodes.*

Suite définie implicitement

Soit f_n une suite de fonction définie sur $\mathbb{R}$ pour tout $n\in\mathbb{N}^*$ tel que :

$$f_n(x)=x^n+x^{n-1}+x-1$$

Question 7 *Montrer que, pour tout $n\in\mathbb{N}^*$, l'équation $f_n(x)=0$ possède une unique solution sur $[0;1]$.*

Notons u_n cette solution.

Question 8 *Écrire un programme Python qui :*

- *définit une fonction `f(x,n)` renvoyant la valeur $f_n(x)$;*
- *définit une fonction `dichotomie(f,n)` renvoyant un valeur à 10^{-3} près de u_n ;*
- *calcule et affiche les 100 premières valeurs approchées de (u_n).*

Question 9 *Conjecturer du comportement de cette suite.*

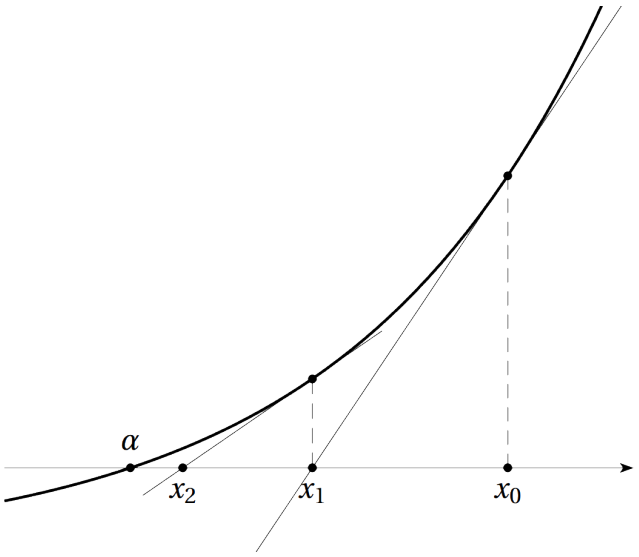
Les délices empoisonnés de la méthode de Newton

Partie A

L'interprétation graphique de la méthode de Newton est donnée ci-contre.

Supposons que les fonctions f et f' soient définies en entête du programme.

Question 1 *Calculer les 20 premières valeurs de la suite générée par la méthode de Newton.*



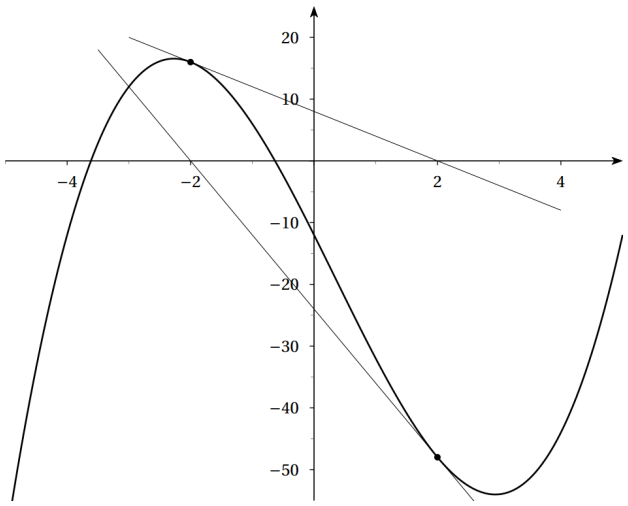
Partie B

Soit la fonction f définie sur $\mathbb{R}$ par : $f(x)=x^3-x^2-20x-12$.

Considérons la suite définie par la méthode de Newton en partant de $x_0=2$

Question 2 *Écrire un programme en Python qui :*

- *définit la fonction f ainsi que sa fonction dérivée que vous noterez fp,*
- *calcule et affiche les valeurs successives de l'algorithme de Newton.*



Exercice : Avec des listes

Lors d'une expérience on mesure un phénomène numérique au cours du temps et on dresse deux listes (de même longueur) :

- `V` : la liste des mesures supposées monotones,
- `T` : la liste des temps (en seconde, dans l'ordre croissant) correspondant à chaque mesure.

Exemple : `T=[0, 2, ...]` et `V=[0.5, 0.55, ...]` signifie que 0.5 a été mesuré à 0s, puis la valeur suivante (0.55) a été prise à 2s etc.

Question 3 *Écrire une fonction* `solution(valeur,V,T)`*, qui renvoie un encadrement du temps pour lequel les mesures encadrent elles-même la valeur.*

Indication :

- privilégiez la dichotomie,
- déterminez les indices `i` et `j` de cet encadrement,
- observez que le test d'arrêt de recherche de ces entiers ne dépend pas d'une précision.

Question 4 *Tester ce programme avec les listes :* `T=[0, 2, 3, 5, 6, 8, 10]` *et* `V=[0.5, 0.55, 0.7, 0.9, 1, 1.5, 1.6]` *avec 0.6 pour la valeur.*

Question 5 *Modifier votre fonction* `solution` *afin de gérer (par un message d'erreur) le cas où il n'y a pas d'encadrement possible car* `valeur` *serait incompatible.*