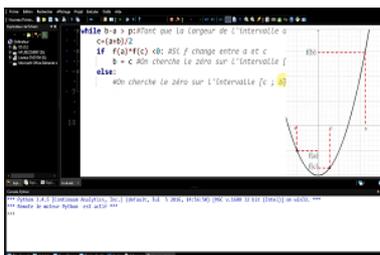


<http://www.lyc-pierre-bourdan.ac-limoges.fr/spip.php?article1565>



Encadrement de la solution \hat{I}_{\pm} d'une équation avec la méthode par dichotomie

- Enseignements disciplinaires - Mathématiques -



Publication date: samedi 8 mai 2021

Copyright © Lycée Pierre Bourdan - Tous droits réservés

Sommaire

- [Le principe et l'algorithme](#)
- [Vidéo : avec Python](#)

Le principe et l'algorithme

La méthode par *dichotomie* (du grec *dikha* "en deux" et *tomein* "couper") ou de *bissection* est un algorithme de recherche du zéro d'une fonction (encadrement de l'unique solution de l'équation $f(x) = 0$).

Elle consiste à réitérer des partages d'un intervalle en deux moitiés, puis à sélectionner celui dans lequel la fonction f est égale à 0.

Si cela est possible, on dégrossit le plus souvent la recherche en se plaçant initialement sur un intervalle $[a ; b]$ où la fonction est **continue, strictement monotone** (croissante ou décroissante) et telle que $f(a) \times f(b) < 0$ (théorème de la bijection).

Pour cela, on calcule le milieu c de l'intervalle $[a ; b]$ ($c = (a + b)/2$), puis on regarde si la solution \hat{x} se trouve dans l'intervalle $[a ; c]$ ou $[c ; b]$.

Si la solution est dans $[a ; c]$, on recommence le procédé dans $[a ; c]$.

Si la solution est dans $[c ; b]$, on recommence le procédé dans $[c ; b]$.

Le cœur de l'algorithme repose sur le test suivant :

<p>si la condition $f(a) \times f(c) < 0$ <i>la solution se trouve entre a et c</i></p> <p>alors c "prend la place de b"</p> <p>sinon c "prend la place de a" <i>la solution se trouve entre c et b.</i></p>

L'algorithme s'arrête quand l'amplitude de l'intervalle de recherche devient inférieure à la précision choisie.

Il renvoie alors en sortie un encadrement $[a ; b]$ de \hat{x} .

(Les valeurs de a et de b sont celles du "moment" ; elles ont évidemment été successivement modifiées par l'algorithme).

Remarque : pour une équation du type $f(x) = k$, il suffit de considérer la fonction g définie par $g(x) = f(x) - k$.

Vidéo : avec Python