

Contrôle individuel : Adaptation langage C

31 Octobre 2008

Recherche de zéros par dichotomie

Soit la fonction f réelle continue de la variable réelle x . Le problème consiste à résoudre l'équation $f(x)=0$ dans un intervalle $[a,b]$. On supposera que cette équation a une et une seule racine dans cet intervalle. On utilisera la méthode de dichotomie qui consiste à déterminer une suite d'intervalles I_n contenant la racine de l'équation et tels que la longueur de I_n soit la moitié de celle de I_{n-1} . L'algorithme est le suivant (on note ε un nombre arbitrairement petit choisi à l'avance : prenons $\varepsilon = 10^{-6}$ et **itermax** un nombre d'itérations à ne pas dépasser) :

Soit $c=(a+b)/2$,

- Si $|f(c)| < \varepsilon$, la racine approché est c
- Si $f(a).f(c) < 0$, on remplace $[a,b]$ par $[a,c]$
- Si $f(a).f(c) > 0$, on remplace $[a,b]$ par $[c,b]$
- Si le processus est réitéré plus de **itermax** fois on arrête le processus et on affiche un message d'erreur.

Et on réitère le processus avec le nouvel intervalle ainsi construit.

Ecrire la fonction **dichot** qui traduit la méthode précédente :

double dichot(**double** a, **double** b, **double** eps, **int** itermax)

qui a pour paramètres

- d'entrée : les bornes de l'intervalle initial **a** et **b**, **itermax** le nombre maximum d'itérations que l'on se fixe (dans le cas de non-convergence) **eps** le nombre ε ,
- de retour : la solution, **c**, approchée trouvée par l'algorithme.

La fonction dont on cherche les zéros répondra à la déclaration suivante :

double fct(**double** x) ;

Nous prendrons comme exemple le polynôme du second degré :

double fct(**double** x)

```
{  
    return x*x - (64./15)*x + 19./5 ;  
}
```

qui possède une unique racine dans l'intervalle [0.5 ;2]