TS

Fiche sur la méthode de dichotomie

I. Principe de base

1°) Problème (cadre et notations)

On considère une fonction $f : [a\,;b]\ (a<b) \to \mathbb{R}$ vérifiant les conditions $\begin{cases} C_1 : \text{continue} \\ C_2 : \text{strictement croissante} \\ C_3 : f(a)<0 \text{ et } f(b)>0 \end{cases}$.

D'après le corollaire du théorème des valeurs intermédiaires, $\exists !\, x_0 \in [a\,;b]\ /\ f(x_0)=0$.

La méthode de dichotomie a pour but de déterminer des encadrements de x_0 aussi précis que l'on désire par une méthode algorithmique.
On déterminer ainsi des valeurs approchées aussi précises que possibles.

2°) Lemme fondamental de comparaison de x_0 avec un réel *c* quelconque de l'intervalle $[a\,;b]$

On considère un réel c quelconque dans l'intervalle $[a\,;b]$.

On cherche à comparer x_0 à c.

Graphique

On a trois cas selon le signe de $f(c)$.

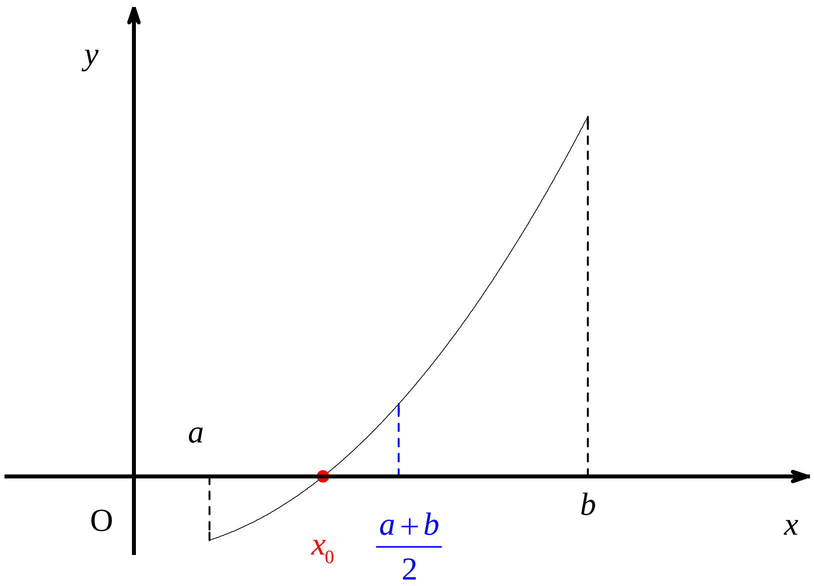
- Si $f(c)<0$, alors $x_0>c$.
- Si $f(c)>0$, alors $x_0<c$.
- Si $f(c)=0$, alors $x_0=c$.

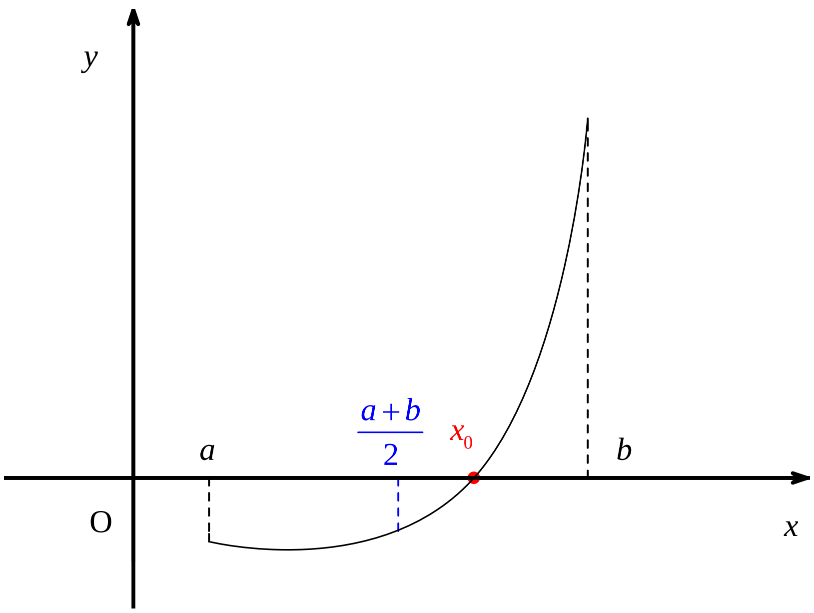
3°) Application du lemme à la dichotomie / début de l'algorithme

On prend pour c le centre de l'intervalle $[a\,;b]$. On a donc $\boxed{c=\dfrac{a+b}{2}}$.

On a trois cas.

Si $f(c)<0$	Si $f(c)>0$	Si $f(c)=0$
$c<x_0\leqslant b$	$a\leqslant x_0<c$	$x_0=c$
$a\leftarrow c$ $b\leftarrow b$	$a\leftarrow a$ $b\leftarrow c$	$a\leftarrow c$ $b\leftarrow c$





II. Exemple de mise en œuvre

On considère la fonction $f : x \mapsto x^2 - 5$ définie sur $\mathbb{R}$.

La fonction f vérifie les trois conditions :

C_1 : f est continue sur $\mathbb{R}$ car c'est une fonction polynôme, donc par restriction sur $\mathbb{R}_+$.

C_2 : f est strictement croissante sur $\mathbb{R}_+$ donc par restriction sur l'intervalle $[2\,;3]$.

C_3 : On a $f(2) = -1$ et $f(3) = 4$ donc 0 est bien une valeur intermédiaire (car 0 est compris entre – 1 et 4).

Par le corollaire du théorème des valeurs intermédiaires, il existe un unique réel x_0 de l'intervalle $[2\,;3]$ tel que $f(x_0) = 0$.

Dans notre cas, on peut facilement trouver que $x_0 = \sqrt{5}$. Nous savons que c'est un nombre irrationnel.

Ce qui va suivre ne présente donc pas un grand intérêt puisque l'on peut aisément obtenir des valeurs approchées de $\sqrt{5}$ grâce à la calculatrice. La méthode présentée ne présente un véritable intérêt que lorsque l'on ne peut pas déterminer la valeur exacte de x_0.

L'intérêt de ce qui va suivre est de trouver des encadrements de $x_0 = \sqrt{5}$ par des décimaux de manière algorithmique en utilisant des opérations.

n	u_n	v_n	c_n	$f(c_n)$	Signe de $f(c_n)$	Encadrement de x_0
0	2	3	2,5	1,25	+	$2 \leqslant x_0 \leqslant 3$
1	2	2,5	2,25	0,0625	+	$2 \leqslant x_0 \leqslant 2{,}5$
2	2	2,25	2,125	– 0,484375	–	$2 \leqslant x_0 \leqslant 2{,}25$
3	2,125	2,25				$2{,}125 \leqslant x_0 \leqslant 2{,}25$

On divise chaque fois par deux l'amplitude de l'encadrement.

On construit ainsi deux suites (u_n) et (v_n) que l'on pourrait définir rigoureusement mais cela n'a pas beaucoup d'intérêt.

On peut démontrer que les suites (u_n) et (v_n) sont respectivement croissante et décroissante.

On peut également démontrer qu'elles convergent vers x_0.

On peut aisément démontrer que $\forall n \in \mathbb{N} \quad v_n - u_n = \dfrac{1}{2^n}$.

Comme $\forall n \in \mathbb{N} \quad u_n \leqslant x_0 \leqslant v_n$, on obtient des encadrements de plus en plus précis de x_0.

On peut aisément réaliser un algorithme puis un programme Python de calcul des termes.

III. Rédaction de l'algorithme et programmation

On considère une fonction $f : [a\,;b] \; (a < b) \to \mathbb{R}$ vérifiant les conditions $\begin{cases} \text{continue} \\ \text{strictement croissante} \\ f(a) < 0 \text{ et } f(b) > 0 \end{cases}$.

1°) La fonction fondamentale (qui change les bornes)

On suppose que la fonction f est rentrée précédemment (vérifiant les hypothèses).

Fonction étape(a, b)

$c \leftarrow \dfrac{a+b}{2}$

Si $f(c) > 0$

$b \leftarrow c$

Sinon

$a \leftarrow c$

FinSi

Renvoyer a, b

On peut dire d'une certaine manière que cette fonction remplace l'intervalle $[a\,;b]$ par l'une des deux « moitiés » d'intervalle.

Il faut savoir la réécrire parfaitement.

2°) Algorithme de dichotomie

Il y a deux options :

- soit on s'oriente vers une boucle bornée (boucle « Pour ») c'est dire que l'on répète la fonction étape un certain nombre de fois fixé à l'avance.

- soit on s'orient vers une boucle non bornée (boucle « Tantque ») c'est-à-dire que l'on va se donner un réel ε au départ (qui va nous servir de précision) et que l'on va obtenir en sortie deux réels a et b tels que $b - a \leqslant \varepsilon$.
On obtient un encadrement de la solution x_0 d'amplitude inférieure ou égale fixée au départ.

Fonction dicho(a, b, n)

 Pour i allant de 1 à n **Faire**

  $a, b = \text{étape}(a, b)$

 FinPour

 Renvoyer a, b

Fonction dicho(a, b, ε)

 Tantque $b - a > \varepsilon$ **Faire**

  $a, b = \text{étape}(a, b)$

 FinTantque

 Renvoyer a, b

3°) Programmation en Python

On reprend la fonction de l'exemple du **II** avec la fonction $f : x \mapsto x^2 - 5$ définie sur $\mathbb{R}$.

Dans la fonction `etape`, on doit avoir $a \leqslant b$ tels que $f(a) \leqslant 0 \leqslant f(b)$ ou $f(b) \leqslant 0 \leqslant f(a)$.

```
def f(x):
    return x*x - 5

def etape(a,b):
    c = (a+b)/2
    if f(c)>0:
        b = c
    else:
        a = c
    return a,b

def dicho(a,b,prec):      # prec signifie précision
    while b-a>prec:
         a,b=etape(a,b)
    return a,b
```

La fonction `dicho` renvoie les réels a et b tels que $b - a \leqslant \text{prec}$.

On sort du script pour l'exécuter et on rentre « `dicho(a,b,prec)` » en remplaçant par les valeurs souhaitées.

Admettons que l'on rentre « `dicho(0,3,0.01)` » avec `a=0`, `b=3` et `prec=0.01`.

La calculatrice renvoie alors : `(2.232421875,2.23828125)`.

On obtient un encadrement de la solution x_0 de l'équation $x^2 - 5$ comprise entre 0 et 3 d'amplitude inférieure ou égale à 0,01.

On a : $2,232421875 \leqslant x_0 \leqslant 2,23828125$.

Comme cette solution est en fait $\sqrt{5}$, on obtient un encadrement de $\sqrt{5}$.

Le 4-3-2024

Programme Python dichotomie

La solution doit préalablement avoir été « localisée » dans un intervalle [a, b].

IV. Adaptation selon la monotonie

1°) Fonction strictement décroissante

On considère une fonction $f : [a\,;b]\ (a<b) \to \mathbb{R}$ vérifiant les conditions $\begin{cases} C_1 : \text{continue} \\ C_2 : \text{strictement décroissante} \\ C_3 : f(a)>0 \text{ et } f(b)<0 \end{cases}$.

Fonction étape(a, b)

$c \leftarrow \dfrac{a+b}{2}$

Si $f(c)>0$

$a \leftarrow c$

Sinon

$b \leftarrow c$

FinSi

Renvoyer a, b

2°) Fonction strictement monotone

On considère une fonction $f : [a\,;b]\ (a<b) \to \mathbb{R}$ vérifiant les conditions

$\begin{cases} C_1 : \text{continue} \\ C_2 : \text{strictement monotone (strictement décroissante ou strictement croissante)} \\ C_3 : f(a)\times f(b)<0 \end{cases}$.

La condition C_3 ($f(a)\times f(b)<0$) exprime que $f(a)$ et $f(b)$ sont de signes contraires.

Fonction étape(a, b)

$c \leftarrow \dfrac{a+b}{2}$

Si $f(a)\times f(c)>0$

$a \leftarrow c$

Sinon

$b \leftarrow c$

FinSi

Renvoyer a, b

2°) Fonction strictement monotone

On considère une fonction $f : [a\,;b]\ (a<b) \to \mathbb{R}$ vérifiant les conditions

$\begin{cases} C_1 : \text{continue} \\ C_2 : \text{strictement monotone (strictement décroissante ou strictement croissante)} \\ C_3 : f(a)\times f(b)<0 \end{cases}$.

3°) Adaptation pour une équation du type $f(x)=k$

On remplace la condition $f(c)>0$ par $f(c)>k$.

4°) Fonction Python `dicho(f,a,b,epsilon)`

```
def dicho(f,a,b,epsilon):
    while b-a > epsilon :
          c = (a+b)/2
          if f(a) * f(c) < 0:
              b = c
          else:
              a = c
    return a,b
```

```
dicho(lambda x : x**2-2,1,2,0.0001)
(1.4141845703125, 1.41424560546875)
```

Le 17-8-2021

Cours de Fabien Pucci PCSI (Lycée Gontran Damas)

```
def dichotomie (f, a, b, epsilon) : # f continue et f(a)□f(t) <=0 avec a<b
      while b – a >= epsilon :
                milieu = (a+b)/2
                if f(a)*f(milieu) <= 0 : # f s’annule dans la première moitié
                    b = milieu
                else :
                     a = milieu # f s’annule dans la deuxième moitié
       return (a+b)/2
```

dicho(0,3,0.01)

Variante importante :

Programme Python :

```
def dichotomie (f,a,b, epsilon ):
    while b-a > epsilon :
          m = (a+b)/2
          if f(m) == 0:
             return m
          elif f(a) * f(m) < 0:
              b = m
          else:
              a = m
    return (a+b)/2
```