Algorithmique et programmation – Terminale spécialité

Chapitre 6 – La méthode de dichotomie

Pr. Elakili · Algorithme · Python · TI-82 · Octobre 2026

Site web : missionmaths360.com

Rappel : le théorème des valeurs intermédiaires et son corollaire (théorème de la bijection)

Si f est **continue** et **strictement monotone** sur $[a\,;b]$, alors pour tout réel k compris entre $f(a)$ et $f(b)$, l'équation $f(x)=k$ admet une **unique** solution α dans $[a\,;b]$. En particulier, si $f(a)$ et $f(b)$ sont de signes contraires, f s'annule une seule fois sur $[a\,;b]$.

Le théorème donne l'**existence** de α, mais pas sa valeur : la dichotomie permet d'en calculer une **valeur approchée**.

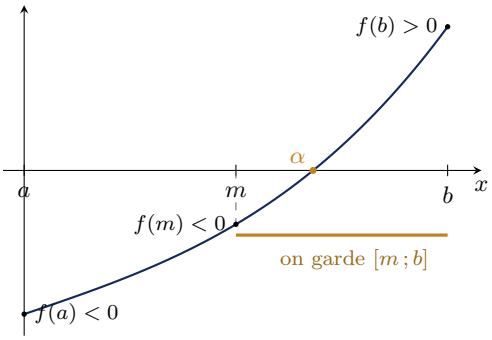
Le principe : couper l'intervalle en deux, et garder la moitié qui contient α

On part de $[a\,;b]$ avec $f(a)$ et $f(b)$ de signes contraires. On calcule le milieu $m=\dfrac{a+b}{2}$:

- si $f(a)$ et $f(m)$ sont de signes contraires, c'est-à-dire $f(a)\times f(m)\leqslant 0$, alors $\alpha\in[a\,;m]$: on remplace b par m ;
- sinon, $\alpha\in[m\,;b]$: on remplace a par m.

On recommence **tant que** l'amplitude $b-a$ est supérieure à la précision ε voulue. À la fin, $a\leqslant\alpha\leqslant b$ avec $b-a\leqslant\varepsilon$.

À chaque étape, l'amplitude est divisée par 2 : après n étapes, elle vaut $\dfrac{b-a}{2^n}$.



$f(x)=x^3+x-1$ sur $[0\,;1]$

Exemple à la main $f(x)=x^3+x-1$ sur $[0\,;1]$, précision $\varepsilon=0{,}01$

f est continue et $f'(x)=3x^2+1>0$, donc f est strictement croissante sur $[0\,;1]$; $f(0)=-1<0$ et $f(1)=1>0$. D'après le théorème de la bijection, $f(x)=0$ a une unique solution $\alpha\in[0\,;1]$.

étape	1	2	3	4	5	6	7
a	0	0,5	0,5	0,625	0,625	0,656 25	0,671 875
b	1	1	0,75	0,75	0,687 5	0,687 5	0,687 5
m	0,5	0,75	0,625	0,687 5	0,656 25	0,671 875	0,679 687 5
signe de $f(m)$	−	+	−	+	−	−	−

Après 7 étapes : $0{,}679\,687\,5\leqslant\alpha\leqslant 0{,}687\,5$, amplitude $\frac{1}{2^7}\approx 0{,}007\,8<0{,}01$. Donc $\alpha\approx 0{,}68$ **à 10^{-2} près** (en réalité $\alpha\approx 0{,}682\,3$).

Pack 1 – l'algorithme de dichotomie (cas général : $f(a)$ et $f(b)$ de signes contraires)

Algorithme

Saisir a, b, ε
Tant que $b-a>\varepsilon$
  $m\leftarrow\dfrac{a+b}{2}$
  Si $f(a)\times f(m)\leqslant 0$ **alors**
    $b\leftarrow m$
  Sinon
    $a\leftarrow m$
  Fin Si
Fin Tant que
Afficher a et b

Python

```
def f(x):
    return x**3 + x - 1
def dichotomie(a, b, eps):
    while b - a > eps:
        m = (a + b) / 2
        if f(a)*f(m) <= 0:
            b = m
        else:
            a = m
    return a, b
```

TI-82

```
:Prompt A,B,E
:TantQue B-A>E
:(A+B)/2→M
:Si Y1(A)*Y1(M)⩽0
:Alors
:M→B
:Sinon
:M→A
:Fin
:Fin
:Disp A,B
```

(avec Y1=X^3+X-1 ; deux Fin : un pour Si, un pour TantQue)

Essai : `dichotomie(0, 1, 0.01)` renvoie `(0.6796875, 0.6875)` : on retrouve le tableau. Avec `eps = 10**-6`, on obtient 0,682 327 … en 20 étapes.

Combien d'étapes ? (une recherche de seuil, chapitre 4)

Après n étapes, l'amplitude vaut $\frac{b-a}{2^n}$. Sur $[0\,;1]$, on cherche le plus petit n tel que $\frac{1}{2^n} \leqslant \varepsilon$, c'est-à-dire $2^n \geqslant \frac{1}{\varepsilon}$:

ε	10^{-1}	10^{-2}	10^{-3}	10^{-6}
étapes	4	7	10	20

car $2^4 = 16$, $2^7 = 128$, $2^{10} = 1\,024$, $2^{20} \approx 10^6$. **À retenir :** environ 3,3 étapes par décimale. C'est sûr mais lent, comparé à la méthode de Héron (chapitre 5) qui double le nombre de décimales à chaque étape.

Pack 2 – l'équation $f(x) = k$ avec f strictement croissante (théorème de la bijection)

On résout $e^x + x = 2$ sur $[0\,;1]$. La fonction $g(x) = e^x + x$ est continue et strictement croissante (somme de deux fonctions strictement croissantes), $g(0) = 1 < 2$ et $g(1) = e+1 \approx 3{,}72 > 2$: l'équation $g(x) = 2$ a une unique solution $\beta \in [0\,;1]$.
Comme g est **croissante**, le test est plus simple : si $g(m) < k$, la solution est **à droite** de m.

Algorithme

Saisir ε
$a \leftarrow 0$; $b \leftarrow 1$
Tant que $b - a > \varepsilon$
  $m \leftarrow \dfrac{a+b}{2}$
  Si $e^m + m < 2$ **alors**
    $a \leftarrow m$
  Sinon
    $b \leftarrow m$
  Fin Si
Fin Tant que
Afficher a et b

Python

```
from math import exp

def g(x):
    return exp(x) + x
def solution(k, eps):
    a, b = 0, 1
    while b - a > eps:
        m = (a + b) / 2
        if g(m) < k:
            a = m
        else:
            b = m
    return a, b
```

TI-82

```
:Prompt E
:0→A
:1→B
:TantQue B-A>E
:(A+B)/2→M
:Si Y2(M)<2
:Alors
:M→A
:Sinon
:M→B
:Fin
:Fin
:Disp A,B
```

(avec Y2=e^(X)+X : 2nde ln)

Résultat : `solution(2, 0.001)` renvoie `(0.4423828125, 0.443359375)` après 10 étapes : $\beta \approx 0{,}443$ à 10^{-3} près.
Si f est strictement décroissante, on inverse le test : si $f(m) > k$, la solution est à droite de m ($a \leftarrow m$).

Vérifier sur la TI-82 sans programme

- **Par la table :** `Y1=X^3+X-1`, puis déf table `DébTable=0.6`, `PasTable=0.01` : $f(0{,}68) < 0 < f(0{,}69)$, donc $0{,}68 < \alpha < 0{,}69$ (c'est la méthode du « balayage »).
- **Par le graphique :** tracer la courbe (graphe), puis 2nde trace (menu CALCULS) → `zéro` : indiquer une borne gauche, une borne droite et une valeur proche ; la TI affiche `X=0.68232780`. Pour $g(x) = 2$, tracer aussi `Y3=2` et utiliser `intersection`.

☞ Les erreurs classiques

- Oublier de **justifier l'existence et l'unicité** de α (continuité, stricte monotonie, valeurs aux bornes) avant de lancer l'algorithme.
- Inverser $a \leftarrow m$ et $b \leftarrow m$: faire un test sur un exemple simple.
- Sur la TI, oublier l'un des deux `Fin`.
- Confondre la précision ε (l'amplitude finale) et le nombre d'étapes.

À vous de jouer (algorithme + Python + TI-82)

1. Montrer que $x^3 - 3x + 1 = 0$ a une unique solution α dans $[0\,;1]$, puis en donner un encadrement d'amplitude 10^{-3} par dichotomie. *(Attention : la fonction est décroissante sur $[0\,;1]$.)*

2. Approcher $\ln 3$ sans la touche ln : montrer que l'équation $e^x = 3$ a une unique solution dans $[1\,;2]$, puis l'approcher à 10^{-3} près.

3. Combien d'étapes faut-il pour obtenir, sur un intervalle de longueur 2, un encadrement d'amplitude 10^{-4} ? Écrire le programme qui calcule ce nombre.

Réponses : 1. $f'(x) = 3x^2 - 3 < 0$ sur $[0\,;1[$, $f(0) = 1$, $f(1) = -1$; après 10 étapes : $0{,}346\,7 \leqslant \alpha \leqslant 0{,}347\,7$ ($\alpha \approx 0{,}347\,3$). **2.** $x \mapsto e^x$ est continue et strictement croissante, $e^1 \approx 2{,}72 < 3 < e^2 \approx 7{,}39$; $1{,}097\,7 \leqslant \alpha \leqslant 1{,}098\,6$, et en effet $\ln 3 \approx 1{,}098\,6$. **3.** $\frac{2}{2^n} \leqslant 10^{-4} \iff 2^n \geqslant 20\,000$: $n = 15$ ($2^{14} = 16\,384$, $2^{15} = 32\,768$).