

# L'algorithme de dichotomie

Henri ROLAND  
Mai 2010

## 1) Le jeu du nombre caché

Albert a choisi un nombre compris entre 1 et 100, Bertrand doit le deviner.

Bertrand fait des propositions et Albert répond "trop grand", "trop petit" ou "gagné". Le jeu s'arrête lorsque Bertrand a trouvé le nombre.

*Faire fonctionner le jeu avec un camarade, archiver le déroulement du jeu dans un tableau sur papier.*

Que faut-il mémoriser ?

Que faut-il archiver dans le tableau ?

On décide d'appeler N le nombre caché  
et R la réponse de Bertrand.

Albert a choisi 63. On obtient le tableau suivant :

N	R	Test	Réponse d'Albert
63	40	$R < N$	Trop petit
	80	$R > N$	Trop grand
	60	$R < N$	Trop petit
	65	$R > N$	Trop grand
	62	$R < N$	Trop petit
	63	$R = N$	Gagné

*On veut maintenant remplacer Albert par un ordinateur et écrire l'algorithme à faire exécuter à l'ordinateur dans ce cas.*

Pour répondre au nombre R proposé par Bertrand on pourra envisager l'instruction conditionnelle :

```
Si R>N alors
    Afficher("Trop grand")
Sinon
    si R<N alors
        Afficher("Trop petit")
    Sinon
        Afficher("Gagné")
```

On obtient ainsi l'algorithme suivant

```
N ← Nombre aléatoire compris entre 1 et 100.
Lire(R)
Tant que R ≠ N Faire
    Si R > N Alors
        Afficher("Trop grand")
    Sinon
        si R < N Alors
            Afficher("Trop petit")
        Sinon
            Afficher("Gagné")
    Si R ≠ N Alors
        Lire(R)
```

Le **Lire(R)** en fin de boucle permet à Bertrand de taper le nombre suivant sauf dans le cas où il a gagné. On aurait préféré avoir le **Lire (R)** en début de boucle, par contre dans ce cas on aurait deux valeurs de R saisies avant le premier **Si...Alors**. Une solution satisfaisante sera d'initialiser R avec une valeur telle que  $R \neq N$  dans tous les cas.

On obtient l'algorithme suivant :

```

N ← Nombre aléatoire compris entre 1 et 100.
R ← 0
Tant que R ≠ N Faire
    Lire(R)
    Si R > N Alors
        Afficher("Trop grand")
    Si R < N Alors
        Afficher("Trop petit")
Afficher("Gagné")

```

Programmation sur TI 82      Programmation en Python 2.6

<pre> :EntAléat(1,100) → N :0 → R :While R ≠ N :Prompt R :If R &gt; N :Then :Disp("TROP GRAND") :End :If R &lt; N :Then :Disp("TROP PETIT") :End :End :Disp("GAGNE") </pre>	<pre> N=randint(1,100) R=0 while R!=N :     R=input('R=')     if R&gt;N:         print 'Trop grand'     if R&lt;N:         print 'Trop petit' print 'Gagné' </pre>
---	--

## 2) Initiation à la dichotomie

*Bertrand joue contre la machine. Archiver le jeu de Bertrand dans un tableau sur papier.*

Quelle technique de jeu peut employer Bertrand pour gagner le plus rapidement possible ?

- Première méthode : déterminer dans quel intervalle [A ; B] se trouve le nombre caché et proposer un entier aléatoire dans cet intervalle.
- Deuxième méthode : déterminer dans quel intervalle [A ; B] se trouve le nombre caché et proposer un entier C proche du milieu de [A;B].

En employant la première méthode, si le nombre caché est 72 :

A	B	C	R=Réponse d'Albert
1	100	53	Trop petit
54	100	80	Trop grand
54	79	57	Trop petit
58	79	65	Trop petit
66	79	78	Trop grand
66	77	71	Trop petit
72	77	77	Trop grand
72	76	72	Gagné

*On veut maintenant remplacer Bertrand par un ordinateur et écrire l'algorithme à faire exécuter à l'ordinateur dans ce cas.*

## Remarque

Les calculatrices programmables "non formelles" ne gèrent que les variables de type numérique et pas les chaînes de caractères. Il faut donc trouver un codage pour la réponse d'Albert.

Pour simplifier, Albert tapera -1 pour "Trop petit" ou 1 pour "Trop grand" ou 0 pour "Gagné".

On obtient l'algorithme suivant :

Première méthode avec un entier aléatoire	Deuxième méthode par dichotomie
<pre>A←1 B←100 R←2 Tant que R≠0 Faire   C←Entier Aléatoire entre A et B   Afficher(C)   Lire(R)   Si R=1 Alors     B←C-1   Si R=-1 Alors     A←C+1</pre>	<pre>A←1 B←100 R←2 Tant que R≠0 Faire   C←PartieEntière(<math>\frac{A+B}{2}</math>)   Afficher(C)   Lire(R)   Si R=1 Alors     B←C-1   Si R=-1 Alors     A←C+1</pre>

### Programmation de la deuxième méthode

#### Programmation sur TI 82

```
:1→A
:100→B
:2→R
:While R≠0
:PartEnt((A+B)/2)→C
:Disp("PROPOSITION=",C)
:Input("REPONSE ?",R)
:If R=1
:Then
:C-1→B
:End
:If R=-1
:Then
:C+1→A
:End
:End
```

#### Programmation en Python 2.6

```
from math import *
A=1
B=100
R=2
while R !=0 :
    C=floor((A+B)/2.)
    print 'Proposition = ',C
    R=input('Réponse ? ')
    if R==1:
        B=C-1
    if R==-1:
        A=C+1
```

## 3) Utilisation de la dichotomie pour résoudre une équation

$f$  est une fonction définie sur l'intervalle  $[a; b]$  et strictement monotone sur  $[a; b]$  . On cherche à résoudre numériquement l'équation  $f(x) = 0$ .

- On peut éliminer tout d'abord les cas où  $f(a) = 0$  ou  $f(b) = 0$ .
- L'existence d'une racine  $\alpha$  sur  $]a; b[$  est subordonnée au fait que  $f(a)$  et  $f(b)$  sont de signes contraires, ce qui équivaut à  $f(a) \times f(b) < 0$ , et que  $f$  est continue sur  $[a ; b]$ .
- Si  $c$  est un réel de l'intervalle  $]a, b[$ , la position de  $\alpha$  par rapport à  $c$  peut être testée par l'instruction suivante :

```

Si  $f(a) \times f(c) \leq 0$  alors
  Rechercher  $\alpha$  sur  $]a;c[$ 
Sinon
  Rechercher  $\alpha$  sur  $]c;b[$ 

```

Dans la pratique on prendra pour  $c$  le milieu de l'intervalle  $[a; b]$ .  
 On itérera le processus jusqu'à obtenir par exemple  $b - a < \varepsilon$  pour  $\varepsilon$  donné.

*Algorithme*

A, B, E et f sont donnés.

```

Si  $f(A) \times f(B) \geq 0$  alors
  Afficher('Pas de racine sur ]A;B[')
Sinon
  Tant que  $B-A \geq E$  Faire
     $C \leftarrow \frac{A+B}{2}$ 
    Si  $f(A) \times f(C) \leq 0$  alors
       $B \leftarrow C$ 
    Sinon
       $A \leftarrow C$ 
  Afficher(A,B)

```

*Ecriture du programme*

Pour le programme sur TI 82 on suppose que la fonction  $(x \mapsto x^3 + x + 1)$  a été préalablement entrée dans la variable Y1.

Programmation sur TI 82

Programmation en Python 2.6

<pre> :Prompt A :Prompt B :Prompt E :If Y1(A)*Y1(B) ≥ 0 :Then :Disp("PAS DE RACINE") :Else :While B-A ≥ E :(A+B)/2 → C :If Y1(A)*Y1(C) ≤ 0 :Then :C → B :Else :C → A :End :End :Disp("A=",A) :Disp("B=",B) </pre>	<pre> def f(x):     return x**3+x+1 A=input('A=') B=input('B=') E=input('Précision =') if f(A)*f(B) &gt;= 0 :     print('pas de racine entre ',A,' et ',B) else:     while B-A &gt;= E :         C=(A+B)/2.         if f(A)*f(C) ≤ 0 :             B=C         else :             A=C     print('Une racine entre ',A, 'et ',B) </pre>
---	--

### Note

Pour une étude plus générale de problèmes liés à la dichotomie vous pouvez consulter les documents suivants disponibles sur le site de l'IREM d'Aix-Marseille : <http://www.irem.univ-mrs.fr/>

- Une méthode pour élaborer des algorithmes itératifs. Auteur : F.Didier.
- Algorithmes et logique au lycée. Auteurs : P.Bouttier, A.Crumière, F.Didier, J-M.Fillia, M.Quatrini, H.Roland.