Recherche de zéros : l'algorithme de dichotomie

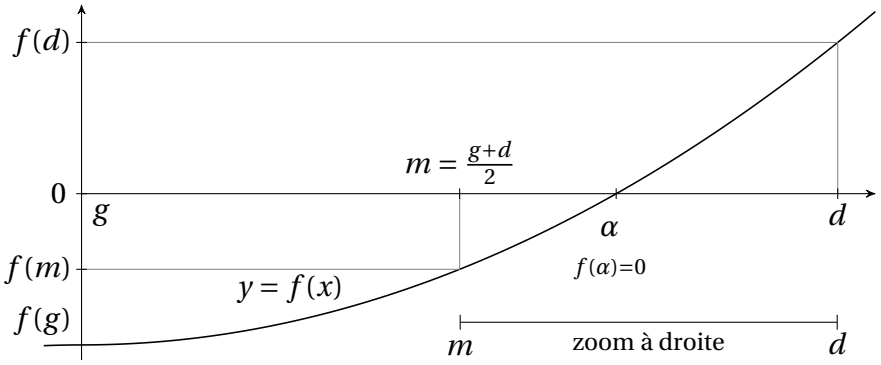
1 Formulation

1.1 Objectif

Proposition 1 (*Théorème des valeurs intermédiaires*)

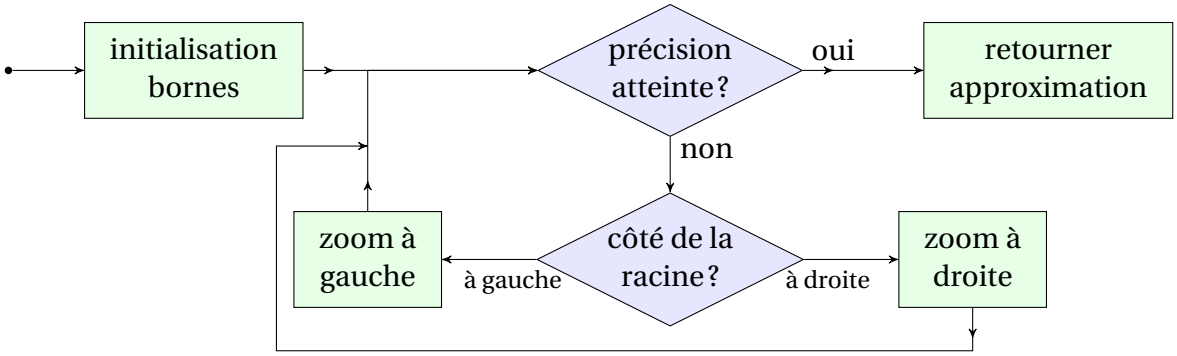
Soit $f : [g;d] \to \mathbb{R}$ une fonction continue qui vérifie : $f(g)f(d) < 0$ *(changement de signes)*. Alors f s'annule sur $[g;d]$: $\quad \exists \alpha \in [g;d],\ f(\alpha) = 0$.
De plus si f est strictement monotone, alors α est unique.

L'algorithme de dichotomie permet de trouver une approximation de ce α *(« la racine »)*.
Pour se fixer les idées, on pense à $f : x \mapsto x^2 - 2$, pour $[g;d] = [0;2]$, et $\alpha = \sqrt{2}$ est unique.



1.2 Organigramme de l'algorithme

À chaque étape de la boucle, on coupe le nouvel intervalle par le milieu.



1.3 Implémentation

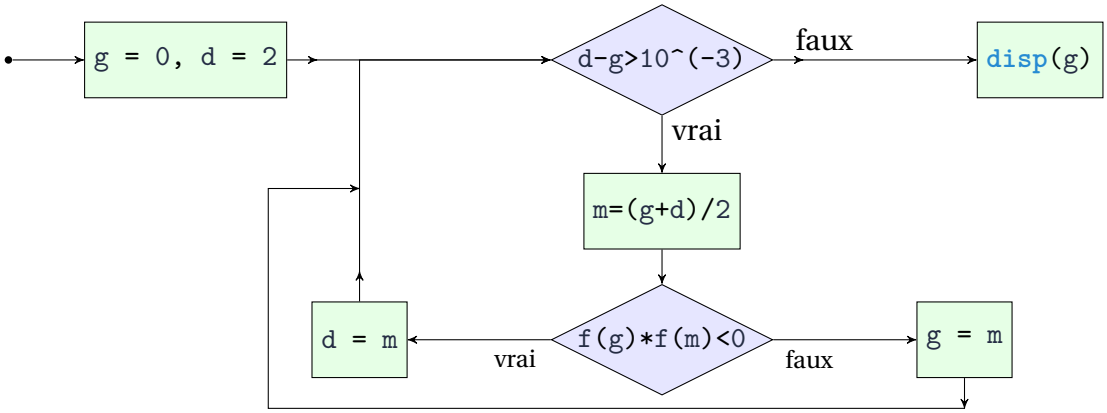
Soit $f : [g;d] \to \mathbb{R}$ une fonction continue telle que $f(g)f(d) < 0$.
On peut approximer un zéro α de f en construisant deux suites comme suit :
Proposition 2 (*Convergence de la méthode de dichotomie*)

- Pour $n \in \mathbb{N}$, on pose $m_n = \frac{g_n + d_n}{2}$ *(le milieu du segment $[g_n;d_n]$)*.
- Si $f(g_n)f(m_n) > 0$, alors : $\begin{cases} g_{n+1} = m_n \\ d_{n+1} = d_n, \end{cases}$ et sinon : $\begin{cases} g_{n+1} = g_n \\ d_{n+1} = m_n \end{cases}$

Alors les suites (g_n) et (d_n) sont adjacentes, et leur limite commune α satisfait $f(\alpha) = 0$.

- **Situer la racine**
  - Si $f(g)f(m) > 0$, alors f change de signes sur $[m\,;d]$, et la racine α est **à droite** de m.
  - Sinon $f(g)f(m) \leqslant 0$, alors f change de signes sur $[g\,;m]$, et α est **à gauche** de m.
- **Zoom**
  - Pour zoomer **à droite**, on remplace $[g\,;d]$ par $[m\,;d]$, soit g par m.
  - Pour zoomer **à gauche**, on remplace $[g\,;d]$ par $[g\,;m]$, soit d par m.

2 En Scilab




```
// Constantes : données du problème
GAUCHE_INIT = 0                              // encadrement initial de la racine
DROITE_INIT = 2
PRECISION   = 10^(-3)                        // précision souhaitée
function y = f(x)                            // fonction étudiée
  y = x.^2 - 2
endfunction

// initialisation
gauche = GAUCHE_INIT
droite = DROITE_INIT

// la boucle de l'algorithme
while ( droite - gauche > PRECISION )        // jusqu'à avoir la bonne précision
  milieu = (gauche+droite) / 2
  if ( f(gauche) * f(milieu) < 0 ) then      // si changement de signe à gauche
    droite = milieu                          //   zoomer à gauche
  else                                       // sinon
    gauche = milieu                          //   zoomer à droite
  end // fin du `if`
end   // fin du `while`

// affichage
disp ("Approximation par défaut de sqrt(2) à 10^(-3) près :")
disp (gauche)                                 // retourne 1.4140625
```