M. Duprez

# TP3 : Minimisation de fonctions non-linéaires

**Exercice 1 : (Un algorithme pour minimiser une fonction d'une variable)**
On cherche à minimiser une fonction $f$ d'une variable réelle $x$

$$\min_{x\in[a,b]} f(x)$$

On propose un algorithme pour résoudre approximativement ce problème (au moins dans le sens de recherche d'un minimum local).

**Algorithme 1 :** section dorée

**Entrées** : Deux nombres $a_0 < b_0$ et la tolérance $tol > 0$
On pose $\tau = \frac{1+\sqrt{5}}{2}$
**pour** $i = 0, 1, \ldots$ **faire**
  $x = a_i + (b_i - a_i)/\tau^2$; $y = a_i + (b_i - a_i)/\tau$
  **si** $f(x) < f(y)$ **alors**
    $a_{i+1} = a_i$; $b_{i+1} = y$
  **sinon**
    $a_{i+1} = x$; $b_{i+1} = b_i$
  **fin**
  **si** $b_{i+1} - a_{i+1} \leq tol$ **alors**
    calculer $c = \frac{a_{i+1}+b_{i+1}}{2}$ et sortir de la boucle
  **fin**
**fin**
**Sorties** : Le nombre $c$ qui donne une estimation d'un minimum local de $f$ avec une erreur plus petite que $tol$

On peut facilement voir (vérifiez !) que cet algorithme converge (si on laisse $i$ tendre vers $\infty$) vers un minimum local de $f$ sur l'intervalle $[a_0, b_0]$ si cet intervalle contient un point $c_0$ tel que $f(c_0) < \min(f(a_0), f(b_0))$. L'idée de preuve consiste à remarquer qu'un tel intervalle contient nécessairement un minimum local de $f$ et l'algorithme construit une succession d'intervalles $[a_0, b_0] \supset [a_1, b_1] \supset \cdots$ de taille de plus en plus petite qui contiennent tous un point $c_i$ tel que $f(c_i) < \min(f(a_i), f(b_i))$.

1. Implémentez l'algorithme de section dorée sous **scilab**.
2. Testez l'algorithme pour quelques fonctions dont vous connaissez les minimums.
3. Visualisez le déroulement de l'algorithme par une petite animation : par exemple, les points $(a_i, f(a_i))$, $(b_i, f(b_i))$ peuvent apparaître l'un après l'autre sur le graphe de $f$.

**Exercice 2 : (algorithme du gradient à pas constant dans l'optimisation sans contraintes)**
On cherche à minimiser une fonction donnée $f : \mathbb{R}^n \to \mathbb{R}$ et on propose l'algorithme suivant dont l'idée est de choisir successivement les points $x^0, x^1, \ldots$ en espérant que cette suite converge vers un minimum local de $f$. Pour construire $x_{i+1}$, on part de $x_i$ dans la direction opposée au gradient de $f$ qui est la direction dans la quelle la fonction est censée de décroître.

**Algorithme 2 :** gradient à pas constant

**Entrées** : $x^0 \in \mathbb{R}^n$, un nombre (le pas) $\rho > 0$ et la tolérance $tol > 0$
**pour** $i = 0, 1, \ldots$ **faire**
  $x^{i+1} = x^i - \rho \nabla f(x^i)$
  **si** $||\nabla f(x^{i+1})|| \leq tol$
    *ou on a perdu l'espoir que l'algorithme aboutisse jamais à quelque chose intéressant*
  **alors**
    poser $x_{\min} = x^{i+1}$, $f_{\min} = f(x_{\min})$ et sortir de la boucle
  **fin**
**fin**
**Sorties** : $x_{\min}$, $f_{\min}$

On verra dans le cours que cet algorithme converge sous certaines hypothèses sur $f$ si $\rho$ est assez petit.

1. Implémentez l'algorithme du gradient à pas constant sous **scilab**.
2. Testez l'algorithme sur quelques fonctions quadratiques, comme, par exemple,

$$f(x_1, x_2) = 2x_1^2 - x_1 x_2 + x_2^2 + 1.$$

   Visualisez le déroulement de l'algorithme en dessinant les lignes de niveau de $f$ et itérations $x^0, x^1, \ldots$ la-dessus.
3. Testez l'algorithme sur les exos du TP n° 2.

**Exercice 3 : (algorithme du gradient à pas optimal dans l'optimisation sans contraintes)**

On cherche toujours à minimiser une fonction donnée $f : \mathbb{R}^n \to \mathbb{R}$ et on améliore l'algorithme de l'exo précédent comme suit : pour construire $x_{i+1}$, on part tojours de $x_i$ dans la direction opposée au gradient de $f$, mais on cherche maintenant à minimiser $f$ dans cette direction. Ainsi on obtient les points $x^0, x^1, \ldots$ tels que $f(x^0) > f(x^1) > \cdots$

**Algorithme 3 :** gradient à pas optimal

**Entrées** : $x^0 \in \mathbb{R}^n$, et la tolérance $tol > 0$
**pour** $i = 0, 1, \ldots$ **faire**
On pose $d^i = \nabla f(x^i)$ et on introduit la fonction $\phi_i(t) = f(x^i - td^i)$ d'une variable $t \in \mathbb{R}^+$.
Soit $\rho_i \geq 0$ le minimum de $\phi_i$. On pose alors
$x^{i+1} = x^i - \rho_i d^i$ **si** $||\nabla f(x^{i+1})|| \leq tol$
*ou on a perdu l'espoir que l'algorithme aboutisse jamais à quelque chose intéressant*
**alors**
poser $x_{\min} = x^{i+1}$, $f_{\min} = f(x_{\min})$ et sortir de la boucle
**fin**
**fin**
**Sorties** : $x_{\min}$, $f_{\min}$

On verra dans le cours que cet algorithme converge sous certaines hypothèses sur $f$.

1. Implémentez l'algorithme du gradient à pas optimal sous **scilab**.
2. Testez l'algorithme sur quelques fonctions quadratiques et sur les exos du TP n° 2.
3. Comparer la vitesse de convergence (le nombre d'itérations nécessaire pour aboutir à une bonne approximation du minimum) de l'algorithme du gradient à pas constant avec celle de l'algorithme du gradient à pas optimal.