TP 1 Optimisation: optimisation locale sans contraintes

L'objectif de cette séance est d'utiliser le logiciel Scilab (ou Matlab, ou Python) afin de comparer différents algorithmes de recherche linéaire pour la minimisation locale d'une fonction définie sur $\mathbb{R}^n$.

Exercice 1

1. On cherche à programmer tout d'abord la méthode du gradient avec une stratégie de type backtracking avec condition d'Armijo. On rappelle que la condition d'Armijo pour un point de départ X_0 et une direction de descente d s'écrit :

$$J(X_0 + \alpha d) \leq J(X_0) + \beta\alpha < d, \nabla J(X_0) > \tag{1}$$

2. Ecrire une fonction Scilab ayant pour arguments J, ∇J, β, α_{init} , τ (paramètres du procédé de backtracking), X_0 (point initial) et N et renvoyant la valeur de X_N.
3. On cherche à minimiser la fonction de Rosenbrock en dimension 2 :

$$J(x, y) = 100(y - x^2)^2 + (x - 1)^2$$

On prend pour cela $\beta = 0.1$, $\alpha_{init} = 1$, $\tau = 0.3$ $X_0 = (0, 1)$.
Représenter graphiquement sur deux figures séparées :
(i) les lignes de niveau de la fonction de Rosenbrock sur $[-1, 2]^2$ et les 100 premières itérations de l'algorithme précédent
(ii) la fonction $N \mapsto J(X_N)$ pour $N \in \{0, ..., 100\}$

Exercice 2

On souhaite remplacer l'algorithme de backtracking par un nouveau procédé de dichotomie où le second critère (en plus du critère d'Armijo) est donné par la condition :

$$J(X_0 + \alpha d) > J(X_0) + \beta_2\alpha < d, \nabla J(X_0) > \tag{2}$$

avec $0 < \beta_2 < \beta$. Ce second critère (condition de Goldstein) permet de sélectionner un pas suffisamment grand. Partant d'un intervalle $[0, \alpha_{max}]$ où α_{max} ne satisfait pas la condition d'Armijo, proposer et implémenter avec Scilab un algorithme permettant de trouver un pas convenable et comparer avec la solution précédente. On pourra prendre $\beta_2 = 0.001$.

Exercice 3 On remplace la condition (2) par la nouvelle condition (dite de Wolfe) :

$$\varphi'(\alpha) > \beta_3\varphi'(0) \tag{3}$$

avec $\varphi(\alpha) = J(X_0 + \alpha d)$ et où $0 < \beta_3 < \beta$. Reprendre l'algorithme de dichotomie construit dans l'exercice 2 et comparer les résultats obtenus avec cette nouvelle méthode (avec $\beta_3 = 0.001$).

Exercice 4 En suivant la même démarche que dans l'exercice 1, écrire une nouvelle fonction Scilab, utilisant la méthode de Newton pour minimiser une fonction J et la tester sur le même exemple (fonction de Rosenbrock). La recherche linéaire est alors supprimée (prendre $\alpha = 1$) et un nouvel argument apparait, en l'occurrence HJ le Hessien de J.

Exercice 5 Même exercice que l'exercice 4, avec la méthode BFGS (à noter que l'argument HJ disparait). Conclure sur l'efficacité comparée des méthode de gradient, Newton et BFGS.