EPFL
ENAC INTER TRANSP-OR
Prof. M. Bierlaire

OPTIMISATION I
GM/EL/CGC/SC/IN
Automne 2012/2013

Serie 3

Afin de pouvoir implémenter à l'aide du logiciel Matlab les différentes méthodes proposées au cours pour résoudre un problème non-linéaire sans contrainte, un programme d'optimisation vous est donné à l'adresse suivante :

`http://transp-or.epfl.ch/courses/optimization2012/exercices.php`

Il comprend six fichiers. Le fichier principal, qui se nomme `optSerie3.m`, utilise les cinq fonctions suivantes :

`bfgs.m`	:	donne la matrice H (**à implementer**),
`taillepas.m`	:	calcule une taille de pas (**à implementer**),
`normGradient.m`	:	fournit un critère d'arrêt (déjà fait),
`visual3d.m`	:	permet la visualisation des résultats (déjà fait),
`f1.m`	:	évalue la fonction objectif, son gradient et son hessien (déjà fait).

La fonction calculée par `f1.m` est :

$$f(x, y) = x^2 + 3y^2.$$

Méthodes quasi-Newton

Supposons à présent que la Hessienne des fonctions-objectifs soit trop coûteuse à calculer ou ne soit plus disponible. Implémenter un algorithme de minimisation qui surmonte ce problème. Utiliser la méthode quasi-Newton proposée par Broyden, Fletcher, Goldfarb et Shanno (BFGS) :

$$H_k = H_{k-1} + \frac{y_{k-1} y_{k-1}^T}{y_{k-1}^T s_{k-1}} - \frac{H_{k-1} s_{k-1} s_{k-1}^T H_{k-1}}{s_{k-1}^T H_{k-1} s_{k-1}}. \tag{1}$$

avec H_0 arbitraire, symétrique définie positive.

A faire

(a) Implementer `bfgs.m` comme indiqué dans (1).

(b) Construire une fonction `taillepas.m` implémentant *la première condition de Wolfe* (décroissance suffisante) pour le calcul de la longueur de pas :

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \beta \nabla f(x_k)^T d_k \quad \text{avec} \quad \beta \in \left]0, 1\right[.$$

(c) Tester l'efficacité de méthode quasi-Newton sur la fonction `f1.m` (avec $H_0 = I$).

(e) Tester la méthode quasi-Newton sur la fonction donnée par :

$$f(x, y) = \frac{1}{2}(y - x^2)^2 + \frac{1}{2}(1 - x)^2.$$

et sur la fonction de Rosenbrock (implémentée lors de la série 1) :

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

(d) Comparer le comportement de cet algorithme en l'initialisant soit avec $H_0 = I$ (matrice identité) soit avec $H_0 = \nabla^2 f(x_0)$ à l'aide des deux fonctions tests définies ci-dessus. Que peut-on en déduire ?

October 23, 2012 – mth/iva