

MASTER MISV 2008–2009

OPTIMISATION ET ALGORITHMIQUE

Projet SCILAB surveillé, durée 2h30

Polycopié de cours et notes de TD/TP autorisés, l'utilisation de tout appareil électronique, hormis les ordinateurs, est interdit.

Problème

On s'intéresse à la minimisation de la fonctionnelle coût : $f(x) = -\log\left(1 - \sum_{i=1}^n x_i^2\right) - \sum_{i=1}^n \log(x_i)$

où : $x = \begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$.

1. Déterminez le domaine de f , $\text{dom}(f)$. Donnez en une représentation graphique pour $n = 2$.
Donnez une interprétation géométrique de ce problème de minimisation.
2. Montrez que minimiser f sur $\text{dom}(f)$ revient à maximiser un polynôme sous contraintes. Donnez ce polynôme et les contraintes explicitement.
En déduire que le minimum de f est atteint en $x_{\min} = \left(\frac{1}{\sqrt{n+2}}, \dots, \frac{1}{\sqrt{n+2}}\right)$.
3. Calculer le gradient $\nabla f(x)$ et la matrice hessienne $\nabla^2 f(x)$.
Écrire les fonctions Scilab : **objectif(x)** qui calcule $f(x)$,
gradient_obj(x) qui calcule $\nabla f(x)$ et **hessienne_obj(x)** qui calcule $H_f(x)$.
4. On pose $\phi(s) = f(x + sd)$, où $x, d \in \mathbb{R}^n$ et $s \in \mathbb{R}_+$. Que vaut $\phi'(s)$?
Écrire la fonction Scilab **d_phi(s)** qui évalue $\phi'(s)$.

On veut comparer les performances de la méthode de descente du gradient, de la méthode de NEWTON et de la méthode du gradient conjugué non linéaire, version FLETCHER-REEVES et POLAK-RIBIÈRE pour des valeurs différentes de la dimension n .

En utilisant la direction de descente d , donnée par **chacune** des méthodes à étudier :

- (i) pour $x \in \text{dom}(f)$, déterminer s_M tel que $x + s_M * d$ est dans $\text{dom}(f)$;
- (ii) déterminer s comme solution de $\phi'(s) = 0$ grâce à :
 $\text{s} = \text{fsolve}(\text{s}_M/2, \text{d_phi})$; où la fonction **d_phi()** a été définie en 4.
 Ceci afin d'avoir un *pas de descente* s *optimal* !

5. Adapter la fonction Scilab de descente de gradient vue en TP : **[x_h, v_h] = methode_gradient(x0)**.
6. Adapter la fonction Scilab vue en TP **[x_h, v_h] = gc_non_lin(x0, flag_bet)** qui minimise la fonction f grâce à la méthode du gradient conjugué non linéaire.
7. Modifier la fonction Scilab vue en TP **[x_h, v_h] = methode_newton(x0)** qui minimise la fonction f grâce à la méthode de NEWTON.
8. Initialiser x_0 de la façon suivante :
 $x_0 = \text{zeros}(n, 1)$; $x_0(1:n-1) = 0.1$; $x_0(n) = 0.8 * \text{sqrt}(1 - \text{sum}(x_0(1:n-1).^2))$
 Comparer les performances des quatre méthodes en traçant la précision du résultat ($-\log_{10} |v - v_{\min}|$) en fonction du nombre d'itérations pour les quatre méthodes ($v_{\min} = f(x_{\min})$).
 Faites croître n de 2 à 30...
 Donnez une estimation du nombre d'opérations par itération.
 Quelle méthode permet d'aller le plus loin ?

À la fin de la séance :

- (1) remettre un rapport avec : les réponses, commentaires
- (2) envoyer les fichiers sources, commandes et les résultats éventuels à
 georges.koepfler@mi.parisdescartes.fr