Université Paris Descartes
UFR de Mathématiques et Informatique
45, rue des Saints-Pères 75270 Paris cedex 06

MASTER MA-IM 2012–2013

Optimisation Algorithmique

Polycopié et notes autorisés. Durée 2h.

Fichiers disponibles :

1. Fiches de TD/TP et corrigé des TP dans le répertoire CORRIGE_TP ;
2. dans le répertoire SCILAB le polycopié Scilab ;
3. les fichiers CC.sci et CC.sci à compléter et qui contiennent une partie de code à utiliser dans l'exercice 2 ;

À remettre :

1. les fichier `CC.sce` et `CC.sci` des fonctions respectivement des commandes Scilab avec votre nom en commentaire ;
2. la copie double sur laquelle vous pouvez expliquer ce que vous avez fait (calculs, formules, graphiques,...).

Exercice I

Soit A une matrice réelle de taille $[n,n]$, symétrique et telle que $\forall x \in \mathbb{R}^n, x^t\, A\, x \geq 0$.

1. Montrer que A peut s'écrire sous la forme $A = U\,\Sigma\,U^t$ avec Σ une matrice diagonale de taille $[n,n]$ et $U^t\,U = U\,U^t = I_n$.
   En déduire une expression pour la matrice carrée réelle X vérifiant $X^2 = A$.

2. Dans le fichier `CC.sci`, écrire une fonction Scilab `function [X] = fnct_exI(A)` qui calcule cette matrice X.

3. Dans le fichier `CC.sce`, écrire des lignes de commande qui testent la fonction `fnct_exI()`.

Exercice II

On considère la fonction $f(x) = \log(e^x + e^{-x}) = \log(2\cosh(x))$, définie pour $x \in \mathbb{R}$.

1. Déterminer $x_{min} = \underset{x\in\mathbb{R}}{\operatorname{argmin}}\, f(x)$. Calculer $f'(x)$ et $f''(x)$.
   Dans le fichier `CC.sci`, compléter les fonctions `objectif()`, `gradient_obj()` et `hess_obj()`.

2. Compléter les lignes d'exécution dans le fichier `CC.sce` afin de pouvoir tester les fonctions `methode_gradient()` et `methode_newton()` fournies dans `CC.sci`.
   Expliquez le test d'arrêt utilisé pour la méthode de Newton.

3. Utiliser `x0 = 1.1` et ensuite `x0 = 1` et comparez la méthode de descente de gradient avec la méthode de Newton. Que constatez vous ? Expliquez.

4. Complétez la fonction `methode_newton()` de façon à déterminer le pas de descente par backtracking si `pas=b` et si `pas=N` on utilise le pas de Newton (*i.e.* $s = 1$, voir le code).

5. Refaites des tests en comparant la vitesse de convergence des trois approches. Commentez les résultats.

Exercice III

On veut minimiser la fonction

$$f(x) = -\log(x_1) - \log(x_2) - \log(1 - x_1 - x_2)\,, \text{ pour } x = (x_1, x_2) \in \mathbb{R}^2\,,$$

en comparant la méthode de descente de gradient, la méthode de Newton, la méthode du gradient conjugué version FLETCHER-REEVES et POLAK-RIBIÈRE.

1. Déterminer $\mathrm{dom}(f)$. Calculer $\nabla f(x)$ et $H_f(x)$.

2. Donnez une interprétation géométrique du problème, en déduire l'existence d'un unique x_{min}. Donnez en la valeur.

3. Copiez dans le fichier `CC.sci` les fonctions `objectif_III()`, `gradient_obj_III()` et `hess_obj_III()` adaptées, ainsi que les fonctions nécessaires pour minimiser f : `methode_gradient_III()`, `methode_newton_III()` et `gc_non_lin_III()`.

4. Dans le fichier `CC.sce`, donnez les lignes de commande qui permettent de calculer x_{min} à chacune des 4 méthodes. On tracera les lignes de niveau de f ainsi que le chemin des $x^{(k)}$ pour chacun des algorithmes. Enfin on représentera une comparaison de la vitesse de convergence pour les 4 méthodes.