CC2 Optimisation numérique 2024-25 : correction

Ex 1) 9 pts

1) (2 pts) C est un ensemble fermé (intersection de fermés car inégalités larges + continuité de Φ_i par convexité). De plus C est borné. Par continuité de J, on obtient l'existence d'un minimum de J sur C (on aurait pu aussi utiliser la coercivité). 1

De plus, C est convexe. En effet, si x et $y \in C$, on a

$$\Phi_i(\lambda x+(1-\lambda)y) \leq \underbrace{\lambda \Phi_i(x)+(1-\lambda)\Phi_i(y)}_{\leq 0}$$

et $\lambda x+(1-\lambda)y \in C$.

Comme J est strictement convexe, on a bien l'unicité du minimum précédent.

2) (2 pts) On considère $K>0$ et on note

$$D_K=\{x\in\mathbb{R}^n \,/\, \Phi_i(x)\leq -\tfrac{1}{K},\ \forall i\}$$

2

On a si $x \in D \setminus D_k$:

$J_\varepsilon(x) \geq J(x) + \varepsilon k$

Comme D_k est un ensemble compact et convexe, en procédant comme en 1), J_ε possède un unique minimum sur D_k. Par k assez grand, ce minimum est aussi l'unique minimum de J_ε sur D.

3) (3 pts)

Avec le raisonnement précédent, on constate que x_ε est localisé dans un compact (du type D_{k_0} avec k_0 fixé) par ε suffisamment petit. Ainsi, quitte à extraire, il existe $\tilde{x} \in D_{k_0}$ tel que $x_\varepsilon \longrightarrow \tilde{x}$ lorsque $\varepsilon \to 0$.

Soit $x \in D$. Par définition $J_\varepsilon(x) \geq J_\varepsilon(x_\varepsilon)$ soit :

$$J_\varepsilon(x) \geq J(x_\varepsilon) - \varepsilon \sum_{i=1}^{m} \frac{1}{\Phi_i(x_\varepsilon)}$$

Par passage à la limite ($\varepsilon \to 0$) sachant que $\tilde{x} \in D_{k_0}$, on a $\Phi_i(\tilde{x}) \leq -\frac{1}{k_0}$, et

3

ainsi :

$$J(x) \geq J(\tilde{x}) \quad \forall x \in D.$$

Comme C est l'adhérence de D cette inégalité est aussi valable sur C et on a bien $\tilde{x} = x^*$.

4) (2pts) On écrit les 2 relations de définition

$$J(x_\varepsilon) - \varepsilon \sum \frac{1}{\Phi_i(x_\varepsilon)} \leq J(x_{\varepsilon'}) - \varepsilon \sum \frac{1}{\Phi_i(x_{\varepsilon'})}$$

et

$$J(x_{\varepsilon'}) - \varepsilon' \sum \frac{1}{\Phi_i(x_{\varepsilon'})} \leq J(x_\varepsilon) - \varepsilon' \sum \frac{1}{\Phi_i(x_\varepsilon)}$$

En effectuant l'opération $\varepsilon' * (1) + \varepsilon * (2)$

on trouve :

$$(\varepsilon' - \varepsilon) J(x_\varepsilon) \leq (\varepsilon' - \varepsilon) J(x_{\varepsilon'})$$

ce qui implique si $0 < \varepsilon' < \varepsilon$:

$$J(x_{\varepsilon'}) \leq J(x_\varepsilon)$$

L'inégalité $J(x^*) \leq J(x_{\varepsilon'})$ provient de la définition de x^*.

Ex 2

9 pts

1) (1,5pts) on a

$$d(x_0, D)^2 = \inf_{y \in D} \|x_0 - y\|^2$$

Il s'agit donc de minimiser

$$J(y) = \|x_0 - y\|^2 = \|y\|^2 - 2\langle x_0, y\rangle + \|x_0\|^2$$

sous les m contraintes

égalité : $Ax = b$

2) (1,5 pts) La fonction J est strictement convexe et coercive sur $\mathbb{R}^n$ ($Id \in S_n^{++}(\mathbb{R})$). Elle possède donc un unique minimum sur D (ensemble fermé et convexe), noté x^*.

3) (3 pts) On a

$$L(y, \lambda) = J(y) + \lambda^T.(Ay - b)$$

$\lambda \in \mathbb{R}^m$

Les conditions KKT s'écrivent

4

ici :

$$\begin{cases} 2x^* - 2x_0 + A^T\lambda^* = 0 \\ Ax^* = b \end{cases}$$

Il vient alors :

$$2b - 2Ax_0 + (A.A^T)\lambda^* = 0$$

Comme A est de rang maximal, la matrice $A.A^T \in S_n^{++}(\mathbb{R})$ et :

$$\lambda^* = -2(A.A^T)^{-1}(b - Ax_0)$$

puis :

$$x^* = x_0 + A(A.A^T)^{-1}(b - Ax_0)$$

4) (1pts) si $m=1$, on a

$$x^* = x_0 + A^T \|A\|^{-2}(b - Ax_0)$$

et $d(x_0, D) = \|x_0 - x^*\|$

$$= \frac{1}{\|A\|^2} \cdot \| A^T \cdot \underbrace{(b - Ax_0)}_{\in \mathbb{R}} \|$$

$$= \frac{1}{\|A\|} |b - Ax_0|$$

5) (2pts) Initialisation : x_0, $\lambda_0 = 0$

$X = x_0$, $\lambda = \lambda_0$

Itération : $(\alpha_1, \alpha_2 : \text{pas})$

$$X \to X - \alpha_1 (2X - 2x_0 + A^T \lambda)$$

$$\lambda \to \lambda + \alpha_2 (AX - b)$$

Ex3 (2pts) 6 pts 5

1) On cherche ici à minimiser la fonction

$$J(x, y) = x^2 + xy$$ sous la contrainte :

$$x^2 y = 2V_0$$

par une méthode de pénalisation (paramètre ρ) résolue par une descente de gradient (recherche linéaire de type "backtracking + Armijo")

2) (1 pt) "rho" représente le paramètre (ou le poids) de la pénalisation choisie. "x" renvoie le résultat de l'optimisation après N itérations

3) (2 pts) On écrit ici

$$x^2 y = 2000 \Rightarrow y = \frac{2000}{x^2}$$

et $J(x,y) = \underbrace{x^2 + \frac{2000}{x}}_{g(x)}$

On calcule $g'(x) = 2x - \frac{2000}{x^2}$

ce qui donne comme variations de g :

x	0		10		$+\infty$
g'		$-$	0	$+$	
g	$+\infty$	$\searrow$	300	$\nearrow$	$+\infty$

Le résultat exact est donc :

$$(x, y) = (10, 20)$$

4) (1 pt) le script canette2 utilise la fonction Hessian : "def Hess(v)" qui calcule la matrice Hessienne

et qui remplace l'itération

x_k en :

```
x_k = x_k - np.inv(HessJ(x_k)) @ GradJ(x_k)
```