Faculté : MI
Département : Maths
Année universitaire : 2022/2023

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

1^{re} année Master
Semestre : 2
Matière : Ana. Matri. Opt.2

TP 2 : Méthode de pénalité quadratique pour un problème d'obstacle.

On souhaite résoudre numériquement un problème d'optimisation avec contraintes suivant :

$$\begin{cases} \min J(u), \\ u \in K = \left\{ u \in \mathbb{R}^N : \varphi_1(u) \leq 0, \ldots, \varphi_p(u) \leq 0 \right\}, \end{cases} \tag{1}$$

où J est la fonction objective et $\varphi_1, \ldots, \varphi_p$ sont les fonctions contraintes.

Un problème d'obstacle

Soit f et g deux fonctions continues données sur $[0,1]$. On souhaite résoudre le problème d'obstacle suivant : trouver $u : [0,1] \to \mathbb{R}$ telle que

$$\left.\begin{array}{r} -u''(x) \geq f(x), \\ u(x) \geq g(x), \\ (-u''(x) - f(x))(u(x) - g(x)) = 0, \end{array}\right\} \quad \text{sur} \quad]0,1[\ \text{ et } u(0) = u(1) = 0. \tag{2}$$

- ☞ La première équation traduit une concavité maximale de la fonction u.
- ☞ La deuxième équation représente l'obstacle : on veut que la solution u soit au dessus de g.
- ☞ La troisième équation traduit le fait que l'on a au moins égalité dans une des deux équations précédentes : soit on résout $-u''(x) = f(x)$, soit $u(x) = g(x)$, et on est sur l'obstacle.

Pour discrétiser le problème (2) par différences finies, on introduit une subdivision uniforme $x_i = ih$ de $[0,1]$, où $h = \frac{1}{N+1}$ désigne le pas en espace du maillage où $i = 0, 1, \ldots, N+1$. Alors, on cherche à résoudre le problème suivant :

$$\left.\begin{array}{r} \frac{-u_{i-1}+2u_i-u_{i+1}}{h^2} \geq f(x_i), \\ u_i \geq g(x_i), \\ \left(\frac{-u_{i-1}+2u_i-u_{i+1}}{h^2} - f(x_i)\right)(u_i - g(x_i)) = 0, \end{array}\right\} \quad \text{pour } i = 1, \ldots, N \text{ et } u_0 = u_{N+1} = 0. \tag{3}$$

On note J_N la fonction définie par :

$$J_N(u) = \frac{1}{2}\langle A_N u, u\rangle - \langle F_N, u\rangle,$$

où A_N et f_N sont donnés par :

$$A_N = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \quad \text{et} \quad F_N = h^2 \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}$$

1. Montrer l'équivalence suivante :

$$\left(\mathbf{u}_N = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \text{ est solution de (3)} \right) \Leftrightarrow \begin{pmatrix} \mathbf{u}_N \text{ minimise } J_N(u) \\ \text{sur } K_N = \{u = (u_i) : u_i \geq g_i, \forall i\} \end{pmatrix} \tag{4}$$

2. Préciser alors les quantités : J, K, φ_i, $i = 1, \ldots, p$, et p du problème (1) dans ce cas.

Méthode de pénalité quadratique

La fonction de pénalité quadratique $q(\gamma, u)$ pour le problème (4) est donnée par :

$$q(\gamma, u) = J_N(u) + \frac{\gamma}{2} \sum_{i=1}^{N} \left(\varphi_i^+(u)\right)^2,$$

où $\gamma > 0$ est le paramètre de pénalité, $\varphi_i(u) = g_i - u_i$ et $\varphi_i^+(u) = \max\{0, \varphi_i(u)\}$ avec $i = 1, \ldots, N$. La méthode de pénalité quadratique est basée sur le problème de minimisation sans contraintes suivant :

$$\begin{cases} \min q(\gamma, u) \\ u \in \mathbb{R}^N. \end{cases}$$

Nous avons l'algorithme suivant :

Algorithme 1 (Algorithme de pénalité quadratique)

1: Pour $k = 1$, initialiser le résidu $r_k = 1$, choisi une valeur initiale $\gamma_k > 0$ pour le paramètre de pénalité, un point de départ u_d^k, un pas $\rho > 0$ et un test d'arrêt ε.
2: Tant que le résidu r_k est plus grand que ε et que le compteur n'est pas trop grand :
3: Calculer $grad^k = \nabla q\left(\gamma_k, u_d^k\right)$.
4: Calculer $u^{k+1} = u_d^k - \rho grad^k$.
5: Calculer $r^{k+1} = \rho \left\| grad^k \right\|$.
6: Poser $k = k+1$, choisi un nouveau paramètre de pénalité $\gamma_{k+1} > \gamma_k$, choisi un nouveau point de départ $u_d^{k+1} = u^{k+1}$ et passer à l'étape 2.

Le gradient de la fonction $q(\gamma, u)$ est donné par :

$$\nabla q(\gamma, u) = A_N u - F_N + \gamma \sum_{i=1}^{N} \varphi_i^+(u) \nabla \varphi_i(u).$$

1. Écrire une fonction **`penalite.m`** qui prend en argument A_N, F_N, $g_N = (g(x_i))_{i=1,\ldots,N}$, γ, un point de départ u_d^0, un pas ρ et un test d'arrêt ε, et qui renvoie le vecteur u_N qui minimise $q(\gamma, u)$ sur $\mathbb{R}^N$ ainsi que le nombre d'itérations effectuées.
2. Créer un script **`scriptTP2_penalite.m`** et tester la fonction **`penalite.m`** pour $N = 50$, $\rho = 10^{-4}$, $\gamma = 10^2, 10^4, 10^6, 10^8, 10^{10}, 10^{12}$, $f(x) = 1$, $g(x) = \max\left(1.5 - 20(x - 0.6)^2, 0\right)$, $u_d^0 = (8, 4, 0, \ldots, 0)$ et $\varepsilon = 10^{-12}$.
3. Afficher à l'aide de la fonction **`fprintf`** le nombre d'itérations ainsi que le temps de calcul pour chaque valeur de γ.
4. Tracer sur une même figure les solutions approchées u_N, ainsi que le graphe de la fonction g.
5. Tester le **`scriptTP2_penalite.m`** pour $f(x) = \pi^2 \sin(\pi x)$, $g(x) = \max\left(1 - 100(x - 0.7)^2, 0\right)$ et $\rho = \frac{2}{\lambda_1 + \lambda_N}$, où λ_1 et λ_N sont respectivement la plus petite et la plus grande valeur propre de A_N.