TP 3 : Optimisation sous contrainte, un problème d'obstacle.

1 Le problème

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

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

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 u est sur l'obstacle.

Comme dans le TP 2, on discrétise le problème 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 et où $i \in \{0, \ldots, N+1\}$. On cherche alors à résoudre le problème suivant :

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

On note à nouveau J_N la fonctionnelle définie par

$$J_N(\mathbf{u}) = \frac{1}{2}(A_N \mathbf{u}, \mathbf{u}) - (\mathbf{f}_N, \mathbf{u}).$$

où A_N et $\mathbf{f}_N$ sont donnés par :

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

On peut alors montrer l'équivalence suivante :

$$\mathbf{u}_N = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} \text{ est solution de (2)}$$

$$\Leftrightarrow \; \mathbf{u}_N \text{ minimise } J_N(\mathbf{u}) \text{ sur } K_N = \{\mathbf{v} = (v_i)_{i=1\ldots N} : v_i \geq g_i \; \forall\, i\}. \tag{3}$$

2 Méthode de gradient projeté

On rappelle l'algorithme du gradient projeté à pas fixe pour minimiser une fonctionnelle $J : \mathbb{R}^N \to \mathbb{R}$ sur un ensemble K, pour un point de départ $\mathbf{u}^0$, un pas ρ et un test d'arrêt ε préalablement définis :

> *Méthode du gradient projeté à pas fixe*
> Initialiser le résidu r^0 à 1 et le compteur k à 0.
> Tant que le résidu est plus grand que ε et que le compteur n'est pas trop grand :
> — calculer la descente $\mathbf{w}^k = -\nabla J(\mathbf{u}^k)$,
> — poser $\mathbf{u}^{k+1} = P_K(\mathbf{u}^k + \rho \mathbf{w}^k)$, où P_K désigne la projection sur K,
> — calculer le résidu : $r^{k+1} = ||\mathbf{u}^{k+1} - \mathbf{u}^k||$,
> — incrémenter le compteur.

1. Montrer que la projection sur K_N est donnée par :

$$P_{K_N}(\mathbf{u}) = (\max(u_i, g_i))_{i=1\dots N}.$$

Ecrire un programme `projK.m` qui prend en argument un point $\mathbf{u}$ et $\mathbf{g}_N = (g(x_i))_{i=1\dots N}$, et qui renvoie $P_{K_N}(\mathbf{u})$.

2. Adapter le programme `gradient_fixe.m` du TP 2 pour écrire un programme `gradient_projete.m` qui prend en argument A_N, $\mathbf{f}_N$, $\mathbf{g}_N$, un point de départ $\mathbf{u}^0$, un pas ρ et un test d'arrêt ε, et qui renvoie le vecteur $\mathbf{u}_N$ qui minimise J_N sur K_N ainsi que le nombre d'itérations effectuées.

3. Créer un script `script_TP3.m` et tester la fonction `gradient_projete.m` pour $f(x) = 1, g(x) = \max(1.5 - 20(x-0.6)^2, 0), N = 2, 5, 20, 50, 100, \varepsilon = 10^-5$, et ρ choisi de façon optimale, c'est-à-dire :

$$\rho = \frac{2}{\lambda_1 + \lambda_N}$$

où λ_1 et λ_N sont respectivement la plus petite et la plus grande valeur propre de A_N. Afficher à l'aide de la fonction `fprintf` le nombre d'itérations ainsi que le temps de calcul pour chaque N. Tracer sur une même figure les solutions approchées $\mathbf{u}_N$, ainsi que le graphe de la fonction g.

4. Reprendre la question précédente pour $f(x) = \pi^2 \sin(\pi x)$.

3 Méthode de pénalisation

La méthode de pénalisation consiste à remplacer le problème de minimisation sous contrainte (3) en une suite de problèmes de minimisation sans contrainte qui converge vers (3).
Soit $\eta > 0$ donné, et J_N^η défini par :

$$J_N^\eta(\mathbf{u}) = J_N(\mathbf{u} + \frac{1}{\eta}\sum_{i=1}^{N}(\max(g_i - u_i, 0))^2.$$

Le minimiseur $\mathbf{u}_N^\eta$ de J_N^η sur $\mathbb{R}^N$ converge vers $\approxeq_N$ quand η tend vers 0.
On admet que le gradient de J_N^η est donné par :

$$\nabla J_N^\eta(\mathbf{u} = A_N\mathbf{u} - \mathbf{f}_N - \left(\frac{2}{\eta}\text{signe}(g_i - u_i)\max(g_i - u_i, 0)\right)_{i=1,\dots,N}$$

où la fonction signe est définie par : $\text{signe}(x) = \begin{cases} -1 & \text{si} \quad x < 0 \\ 0 & \text{si} \quad x = 0 \\ 1 & \text{si} \quad x > 0 \end{cases}$

5. Adapter la fonction `gradient_fixe.m` du TP2 pour écrire une fonction `penalisation.m` qui prend en argument $A_N, \mathbf{f}_N, \mathbf{g}_N = (g(x_i))_{i=1\dots N}, \eta$, un point de départ $\mathbf{u}^0$, un pas ρ et un test d'arrêt ε et qui renvoie le vecteur $\mathbf{u}_N$ qui minimise J_N^η sur $\mathbb{R}^N$ ainsi que le nombre d'itérations effectuées.

6. Tester cette fonction pour les valeurs numériques de la question **3.**, en prenant :
$N = 50, \rho = 0.1$ et $\eta = 10^5, 10^4, 10^3, 10^2, 10, 1, 0.1$. Afficher à l'aide de la fonction `fprintf` le nombre d'itérations ainsi que le temps de calcul pour chaque η. Tracer sur une même figure les iso-solutions approchées $\mathbf{u}_N^\eta$, ainsi que le graphe de la fonction g et le graphe de la solution de $-u''(x) = 1$. Que constatez-vous pour η grand ? Pour η petit ?