

TP 2 Optimisation: algorithme de pénalisation

L'objectif de cette séance est d'utiliser Python (ou Matlab/Scilab) afin de comparer dans différents contextes la recherche du minimum d'une fonction $J : \mathbb{R}^n \rightarrow \mathbb{R}$. sous une contrainte égalité $c_E(x) = 0$.

Exercice 1

On suppose que le gradient de J et de c_E est connu et facile à calculer.

1. On cherche à programmer la méthode de pénalisation pour résoudre le problème. Ecrire une fonction Python ayant pour arguments $J, \nabla J, c_E, \nabla c_E, X_0, \beta, \alpha_{init}, \tau$ (paramètre de l'algorithme gradient+ backtracking), ρ (paramètres de pénalisation), N et renvoyant la valeur de X_N après N itérations.
2. Appliquer l'algorithme précédent sur le problème de la canette (V0=330cl) en testant différentes valeurs de ρ .

Exercice 2

On suppose que le gradient de J et de c_E est inconnu.

1. Utiliser la méthode 'pattern search+pénalisation' (ou Nelder-Mead +pénalisation) pour résoudre le problème de minimisation précédent. On écrira une fonction Python avec pour arguments $J, c_E, X_0, \alpha_{init}, \rho$ (paramètre de pénalisation), N . On pourra choisir pour famille de directions dans la méthode pattern search la famille à $2n$ éléments :

$$\mathcal{D} = \{e_1, -e_1, \dots, e_n, -e_n\}$$

2. Comparer les résultats obtenus sur l'exemple de la canette.