

TP 3 Optimisation: optimisation sans gradient

L'objectif de cette séance est d'utiliser Python (ou Matlab/Scilab) afin de comparer différents algorithmes de recherche du minimum d'une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$. sans l'utilisation du gradient de celle-ci.

Exercice 1

Ecrire une fonction Python ayant pour arguments f , X_0 et N et renvoyant la valeur de X_N après N itérations de la méthode 'pattern search'. On pourra choisir pour famille de directions la famille à $2n$ éléments $\{e_1, -e_1, \dots, e_n, -e_n\}$.

Appliquer l'algorithme 'pattern search' à l'exemple de la fonction de Rosenbrock à 2 variables (avec une représentation graphique) et à n variables :

$$f(x_1, \dots, x_n) = 100(x_n - x_{n-1}^2)^2 + (x_{n-1} - 1)^2 + \dots + (x_1 - 1)^2$$

Exercice 2

Ecrire une fonction Python ayant pour arguments f , X_0 et N et renvoyant la valeur de X_N après N itérations de la méthode de Nelder Mead (telle que décrite dans le cours).

Appliquer l'algorithme de Nelder Mead sur le même exemple que l'exercice 1.

Comparer la vitesse de convergence sur l'exemple de la fonction de Rosenbrock en traçant l'erreur commise en fonction du nombre d'appels à la fonction f .