

## TP 2: optimisation sous contraintes

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

### Exercice 1

1. On cherche à programmer tout d'abord la méthode d'Uzawa-Hurwicz pour résoudre le problème général de minimisation d'une fonction  $J$  sous la contrainte  $c_E(x) = 0$ .

Ecrire une fonction Python ayant pour arguments  $J, \nabla J, c_E, \nabla c_E, X_0, \lambda_0$  (point initial),  $\alpha_1$  et  $\alpha_2$  (pas) et  $N$  et renvoyant la valeur de  $X_N$  et de  $\lambda_N$  après  $N$  itérations.

2. On suppose que  $J$  est une fonction quadratique avec  $A$  matrice du Laplacien discrétisé (2 sur la diagonale,  $-1$  sur la première sur-et sous-diagonale, 0 ailleurs),  $b$  vecteur formé de 1, et la contrainte s'écrit

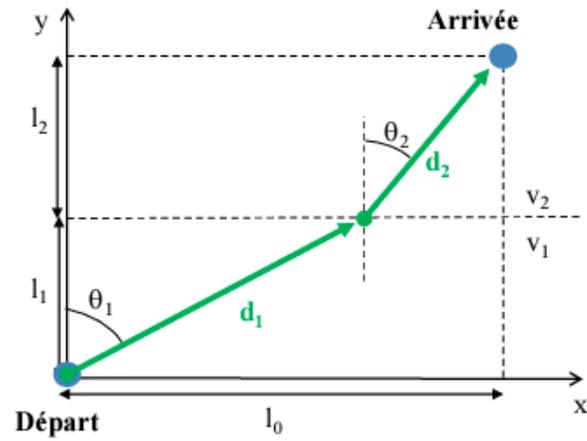
$$\sum_{i=1}^n x_i = 1 \text{ (pour } n = 10 \text{ par exemple) .}$$

Appliquer l'algorithme d'Uzawa sur cet exemple.

### Exercice 2

Appliquer l'algorithme précédent pour résoudre le problème du sauveteur qu'on rappelle ici :

"un sauveteur doit secourir un baigneur en difficulté. Celui-ci se situe à  $l_2$  mètres du rivage tandis que le sauveteur est à  $l_1$  mètres de l'eau et se trouve à une distance longitudinale de  $l_0$  mètres du nageur. Sachant que le sauveteur court à la vitesse  $v_1$  et nage à la vitesse  $v_2$ , trouver de quelle façon il doit secourir le nageur."



### Exercice 3

1. En suivant la même démarche que dans l'exercice 1, écrire une nouvelle fonction Python, utilisant l'algorithme de pénalisation (avec la norme 2). Les variables duales disparaissent dans ce cas des arguments d'entrée et un nouveau paramètre  $\rho$  (paramètre de pénalisation) est ajouté.
2. Comparer les résultats de l'algorithme de pénalisation avec celui d'Uzawa sur l'exemple de l'exercice 1 et de l'exercice 2.