
TP P.L. : La méthode du simplexe

Dans ce TP, on considère uniquement des programmes linéaires (min ou max) écrits sous forme canonique pure et dont le second membre B vérifie $B \geq 0$.

Exercice 1. *Ecrire un programme principal nommé :*

main_simplex.m

dans lequel vous entrez :

- la matrice A des contraintes d'un problème de maximisation sous forme canonique,
- le second membre B des contraintes,
- le vecteur c de la fonction objectif.

Vous assemblez ensuite la matrice initiale de la méthode des tableaux.

Exercice 2. *Faire une fonction matlab*

$[X_{opt}, Val_{foncobjectif}] = simplex_pbmax(\mathbf{mat}).$

Elle prend comme paramètre d'entrée \mathbf{mat} la matrice construite à l'initialisation de la méthode des tableaux. Elle a comme paramètres de sortie, la solution optimale du problème de maximisation associé (rangée dans le bon ordre) ainsi que la valeur de la fonction objectif (avec le bon signe). Elle permet la résolution du problème grâce à la méthode des tableaux.

Exercice 3. 1. *Ajouter à votre programme principal la ligne de commande permettant l'appel à la fonction de l'exercice 2.*
 2. *Tester votre programme sur plusieurs exemples pris dans vos TDs.*

Exercice 4. *Faire une fonction matlab*

$[X_{opt}, Val_{foncobjectif}] = simplex_pbmin(\mathbf{mat}).$

Elle prend comme paramètre d'entrée \mathbf{mat} la matrice construite à l'initialisation de la méthode des tableaux. Elle a comme paramètres de sortie, la solution optimale du problème de minimisation associé (rangée dans le bon ordre) ainsi que la valeur de la fonction objectif (avec le bon signe). Elle permet la résolution du problème grâce à la méthode des tableaux.

Exercice 5. *Ajouter dans votre programme principal le cas de l'exemple du chapitre de cours concernant la phase I. Vous devez entrer les données du problème, trouver une solution de base réalisable initiale, extraire du problème auxiliaire résolu la matrice initiale de la méthode des tableaux (vous pouvez pour cela ajouter à votre fonction `simplex_pbmin` une variable de sortie) et appliquer votre fonction de résolution d'un problème de maximisation.*

Remarque : Matlab possède une fonction "linprog" permettant la résolution d'un programme linéaire. En voici un exemple d'utilisation (tiré des exemples de TD). Celle-ci pourra être utilisée pour vérifier les résultats de vos codes en cas de doute.

```
A = [0, 3; 1.5, 4; 2, 3; 3, 0];
B = [39; 60; 57; 57];
C = -[1700; 3200]; % Attention, linprog cherche un min!
options = optimset('Algorithm','interior-point-legacy');
Aeq=[]; beq=[]; % pas de contraintes d'égalités
lb=zeros(2,1);
ub=[];
x0=[];
% bornes inférieures = 0
% pas de borne supérieure
% on ne fournit pas de solution réalisable au départ
[X, fval] = linprog(C, A, B, Aeq, beq, lb, ub, x0, options)
```