# Optimisation linéaire et convexité
# TP2

10 février 2016

Maxime Chupin, bureau 16-26-333, `chupin@ann.jussieu.fr`.

L'objectif de cette séance est double. Il s'agira dans un premier temps de revenir sur la méthode du pivot de Gauss-Jordan en particulier sur l'interprétation vectorielle de cette dernière. Dans le cadre de la résolution de problèmes d'optimisation linéaire, ce formalisme est en effet très utile. Dans un second temps, la résolution graphique de problèmes d'optimisation linéaire en dimension 2 sera traitée. Cette méthode de résolution s'étend bien sûr difficilement aux problèmes posés en dimensions supérieures mais fournit néanmoins l'intuition géométrique des différentes situations rencontrées.

## 1 Pivot de Gauss-Jordan, suite

On considère une matrice $A \in \mathcal{M}_{p,n}(\mathbf{R})$. On notera $L_i$, $1 \leqslant i \leqslant p$, les lignes de la matrice $A$, et on adoptera également la représentation suivante : $A = (V_1, \ldots, V_n)$ où $1 \leqslant k \leqslant n$, $V_k$ désigne les coordonnées du vecteur éponyme dans la base canonique $\mathcal{B} = \{\epsilon_1, \cdots, \epsilon_p\}$ de $\mathbf{R}^p$. Lorsqu'on a choisi $a_{jm}$ pour pivot, la méthode de Gauss-Jordan consiste en la série de transformations suivantes , pour tout $1 \leqslant i \leqslant p$ :

1. si $i = j$ :
$$L_i \leftarrow L_i / a_{im}$$
2. si $i \neq j$ :
$$L_i \leftarrow L_i - \frac{a_{im}}{a_{jm}} L_j$$

La matrice obtenue contient les coordonnées des $\{V_k\}_{1 \leqslant k \leqslant n}$ dans la base :

$$\mathcal{B}' = \{\epsilon_1, \cdots, \epsilon_{j-1}, V_m, \epsilon_{j+1}, \cdots, \epsilon_p\}$$

de $\mathbf{R}^p$ dite *base voisine* de $\mathcal{B}$.

**Exercice 1**: *Interprétation vectorielle*

1. Écrire une fonction Python `pivot` qui à une matrice $A \in \mathcal{M}_{p,n}(\mathbf{R})$ associe la matrice $B$, de mêmes dimensions, contenant les coordonnées des vecteurs $V_k$ dans la base $\mathcal{B}' = \{\epsilon_1, \cdots, \epsilon_{j-1}, V_m, \epsilon_{j+1}, \cdots, \epsilon_p\}$ de $\mathbf{R}^p$. Cette fonction est de la forme :

```
function pivot(A,j,m):
    ...
    return B
```

*__Remarque__ : il faut bien sûr s'assurer avant d'appliquer cette fonction que $\mathcal{B}'$ définit bien une nouvelle base de $\mathbb{R}^p$ ; ce qui n'a rien d'évident lorsqu'on considère une matrice quelconque $A \in \mathcal{M}_{p,n}(\mathbf{R})$.*

2. Soit la matrice $A = (V_1, V_2, V_3, V_4)$

$$A = \begin{pmatrix} 3 & 4 & -2 & 1 \\ 1 & -1 & 0 & 1 \\ -2 & 0 & 8 & 0 \end{pmatrix}$$

— Calculer à l'aide de la fonction `np.linalg.matrix_rank()` le rang de cette matrice.
— En utilisant la fonction `pivot`, calculer la matrice $B$ des coordonnées des $\{V_k\}_{1\leqslant k\leqslant 4}$. dans la base $\{\epsilon_1, \epsilon_2, V_3\}$.
— Définir la matrice $P = (\epsilon_1, \epsilon_2, V_3)$, matrice de passage de la base $\{\epsilon_1, \epsilon_2, \epsilon_3\}$ vers la base $\{\epsilon_1, \epsilon_2, V_3\}$.
— Comparer $A$ et $C = PB$. Conclusion.
— Pourquoi ne peut-on faire entrer $V_2$ ou $V_4$ dans la base à la place de $\epsilon_3$ ?

**Exercice 2**: Application à la recherche *algébrique* de solutions de base.
On s'intéresse au problème de programmation linéaire dont la forme initiale est la suivante.

$$\text{maximiser } Z(x_1, x_2, x_3) = 20x_1 + 15x_2 + 25x_3,$$

sous les contraintes

$$\left\{ \begin{array}{ccccccc} 2x_1 & + & & & 5x_3 & \leqslant & 5 \\ 2x_1 & + & 2x_2 & + & 3x_3 & \leqslant & 21 \\ x_1 & + & x_2 & + & 4x_3 & \leqslant & 10 \\ & & & & x_1, x_2, x_3 & \geqslant & 0. \end{array} \right. \tag{1}$$

1. Questions de cours
— En introduisant les variables d'écart $\epsilon_1, \epsilon_2, \epsilon_3$, écrire la formulation standard du problème.
— Écrire le problème obtenu après introduction des variables d'écart sous forme matricielle $AX = b$, où $A \in \mathcal{M}_{3,6}(\mathbf{R})$ et $b \in \mathbf{R}^3$.
— Donner une solution de base du système des contraintes. Est-elle réalisable ? Commentaires.
2. Exploration des solutions de base
— Définir avec Python la matrice $M = (A, b)$. *C'est à cette matrice que l'on va appliquer la méthode du pivot de Gauss-Jordan.*
— La matrice $M$ ainsi définie correspond au tableau 1 ci-dessous.

| | $V_1$ | $V_2$ | $V_3$ | $\epsilon_1$ | $\epsilon_2$ | $\epsilon_3$ | $d$ |
|---|---|---|---|---|---|---|---|
| $\epsilon_1$ | 2 | 0 | 5 | 1 | 0 | 0 | 5 |
| $\epsilon_2$ | 2 | 2 | 3 | 0 | 1 | 0 | 21 |
| $\epsilon_3$ | 1 | 1 | 4 | 0 | 0 | 1 | 10 |

Table 1 – Tableau associé au problème (1)

En utilisant la fonction `pivot`, définie dans l'exercice 1, calculer la décomposition des vecteurs $V_1, V_2, V_3, \epsilon_1, \epsilon_2, \epsilon_3$ dans les bases successives :

$$\{V_3, \epsilon_2, \epsilon_3\},\ \{V_3, \epsilon_2, V_2\},\ \{V_3, \epsilon_3, V_2\},\ \{V_3, V_1, V_2\},\ \text{et } \{\epsilon_2, V_1, V_2\},$$

qui sont *dans cet ordre voisine l'une de l'autre.*
— Quelles sont les solutions de base associées à chacune de ces bases. Lesquelles sont réalisables ?

# 2 Résolution géométrique en dimension 2

Lorsqu'on travaille en dimension 2 et que le nombre de contraintes est restreint, on peut aisément résoudre un problème de programmation linéaire à l'aide d'une représentation géométrique. En effet, à chaque contrainte *initiale* de type :

$$(C_i) : a_i x + b_i y \leqslant (\geqslant) c_i,$$

on associe un sous-ensemble de $\mathbf{R}_+^2$ défini par

$$C_i = \left\{(x, y) \in \mathbb{R}_+^2 \mid y \leqslant (\geqslant) \frac{c_i}{b_i} - \frac{a_i}{b_i} x \right\}.$$

L'ensemble $C = \cap_i C_i$ est convexe. C'est l'ensemble des points pour lesquels toutes les contraintes *initiales* sont vérifiées. La compacité de $C$ et la forme de $Z$ décident alors de l'existence et la nature des solutions. La faiblesse de cette approche tient évidemment à ce que lorsque la dimension et le nombre de contraintes croissent, $C$ n'est plus représentable.

**Exercice 3**: *Premier exemple*

On s'intéresse au problème de programmation linéaire dont la forme initiale est la suivante.

$$\text{maximiser } Z(x, y) = 2x + y,$$

sous les contraintes

$$\left\{\begin{array}{ccccc} x & + & 2y & \leqslant & 8 \\ x & + & y & \leqslant & 5 \\ 9x & + & 4y & \leqslant & 36 \\ & & x, y & \geqslant & 0. \end{array}\right. \tag{2}$$

1. Donner une solution de base réalisable.
2. À l'aide de la fonction `plot` de la librairie `matplotlib`, représenter le convexe $C$ associé aux contrainte (2). Que peut-on en conclure quant à l'existence de solutions ?
3. À l'aide de la représentation avec `matplotlib` des lignes de niveaux de la fonction de coût $Z$, résoudre le problème.
4. Que se passe-t-il si l'on cherche à maximiser la fonction de coût $Z_2(x, y) = x + y$ sous les mêmes contraintes ?
5. On suppose, plus généralement, que la fonction de coût $Z_a(x, y) = ax + y$ dépend d'un paramètre $a$. Décrire la solution du problème de programmation linéaire suivant les valeurs du paramètre $a$.

**Exercice 4**: Différentes situations en dimension 2.

Résoudre graphiquement les problèmes de programmation linéaires suivants. Répondre dans chaque cas aux questions suivantes :

1. Représenter à l'aide de `matlplotlib` le convexe associé aux contraintes.
2. Existence et nature des solutions.

**Problème 1**

$$\text{Maximiser } Z_1(x, y) = 2x + y,$$
$$\left\{\begin{array}{ccccc} x & + & y & \geqslant & 4 \\ x & + & 2y & \leqslant & 5 \\ x & & & \geqslant & 2 \\ & & x, y & \geqslant & 0. \end{array}\right. \tag{3}$$

**Problème 2**

$$\text{Maximiser } Z_2(x, y) = 2x - 3y,$$
$$\left\{\begin{array}{ccccc} 3x & + & 2y & \leqslant & 6 \\ x & - & 2y & \geqslant & 5 \\ 2x & - & y & \geqslant & 6 \\ & & x, y & \geqslant & 0. \end{array}\right. \tag{4}$$

**Problème 3**

$$
\text{Maximiser } Z_3(x,y) = 30x + 20y,
$$

$$
\left\{\begin{array}{ccccc}
x & - & y & \leqslant & -2 \\
x & - & 4y & \leqslant & -2 \\
2x & + & y & \geqslant & 5 \\
5x & + & 7y & \leqslant & 44 \\
 & & x, y & \geqslant & 0.
\end{array}\right. \tag{5}
$$

**Exercice 5**: Solutions de base réalisable et sommet du convexe $C$.

On s'intéresse au problème (2) de l'exercice 3, pour la fonction de coût $Z$.

1. Mettre le problème sous forme standard et introduire les variables $e_1$, $e_2$, $e_3$.
2. Encore en utilisant la fonction `pivot`, parcourir *dans cet ordre* les bases voisines suivantes :

$$
\begin{gathered}
\{V_2, \epsilon_2, \epsilon_3\},\ \{V_2, \epsilon_1, \epsilon_3\},\ \{V_2, \epsilon_1, \epsilon_2\} \\
\{V_1, \epsilon_1, \epsilon_2\},\ \{V_1, \epsilon_1, \epsilon_3\}, \\
\{V_1, V_2, \epsilon_3\},\ \{V_1, V_2, \epsilon_2\},\quad \{V_1, V_2, \epsilon_1\}.
\end{gathered}
$$

3. Relever les solutions de base réalisables.
4. Comparer avec les sommets du convexe des contraintes $C$, tracé dans l'exercice 3. Conclusion ?