Michel TAIX

30/08

Introduction à l'optimisation Numérique 1

⚠ Revoir le Python ! (voir le MOOC de la plateforme FUN)

Partie 1 - Concepts et outils de base pour l'optimisation

1.1 Modélisation mathématique : (diapo 9)

Exemple 1 :

$$P = \begin{pmatrix} A \\ B \end{pmatrix}$$

Le modèle que tout point doit vérifier $y_i = a x_i + b$

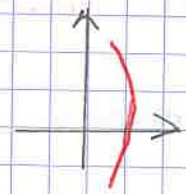
L'erreur est définie par $y_i - a x_i + b = 0$

On souhaite minimiser $\underbrace{\min_{(a;b)}}_{\text{variables de décision}} \sum_{i=1}^{N} (y_i - a x_i - b)^2$

=> il s'agit d'un problème d'optimisation sans contraintes non linéaire

↳ résolu par les Moindres Carrés.

Remarque : Une contrainte aurait pu être se limiter à une partie du plan

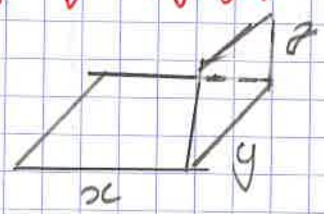


Exemple 2 : Les variables de décisions sont $\underset{\text{longueur}}{x}; \underset{\text{largeur}}{y}; z$ ↑ hauteur

On veut minimiser :

$$\min_{(x;y;z)} \underbrace{\left[N_1 (2xy + 2xz) + N_2 (x \cdot y) + N_3 (x \cdot y) \right]}_{f(\underline{x}) = f(x;y;z) \text{ non linéaire}}$$

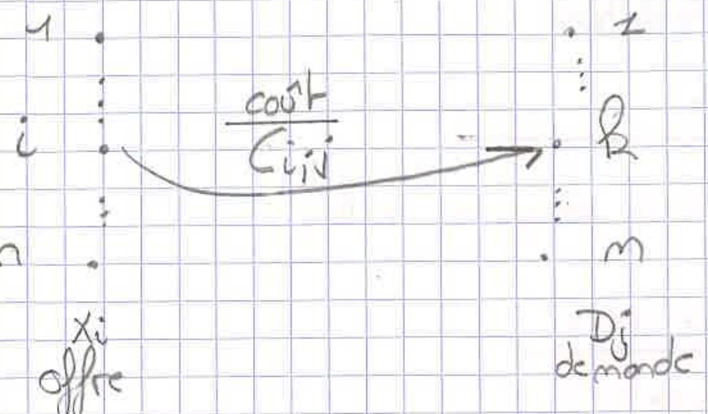
Les contraintes sont $\begin{cases} x \cdot y \cdot z = 1500 \\ y = 2z \end{cases}$



=> c'est un problème d'optimisation sous contraintes non linéaire

Exemple 3 :

dépôt n : point de vente m :



⚠ cahier des charges flou -> faut-il satisfaire toute la demande ? -> autre ? } les contraintes en découlent

Quantité $Q_i = X_i \times$ Pour aller de i vers j on choisit d'affecter une quantité $x_{i,j}$

Par exemple, $c_{11} x_{11} + c_{12} x_{12} + c_{1m} x_{1m} + \dots +$

La fonction est une double somme $\boxed{\min_{(x_{ij})} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \cdot x_{ij}}$

=> il faut prendre en compte des contraintes dépendant de X_i et D_j (les stocks maxi)

choix pour vider les stocks à moitié au moins

Contraintes $\left\{ \underbrace{\frac{X_i}{2} \leq \sum_{j=1}^{m} x_{i,j}}_{\text{fonction linéaire}} \right.$

On choisit de vider la totalité du dépôt $\left\{ X_i = \sum_{j=1}^{m} x_{i,j} \right.$ pour $i = 1, \dots, n$

(ici nous sommes en présence de n contraintes égalité)

ET

On choisit de satisfaire la totalité de la demande des points de vente $\left\{ D_j = \sum x_{i,j} \right.$ pour $j = 1, \dots, m$

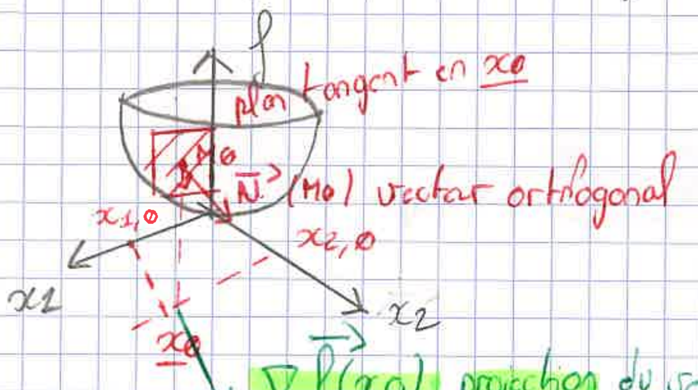
=> C'est un problème d'optimisation linéaire avec contraintes linéaires

1.3. Rappel et compléments diapo 10

Soit une fonction $\underline{x} \longmapsto f(\underline{x})$, $\mathbb{R}^n \longrightarrow \mathbb{R}$

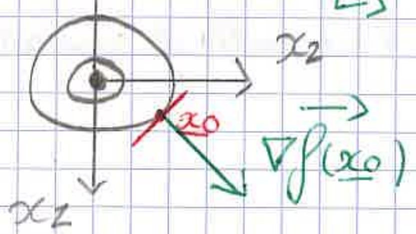
Par exemple la fonction non linéaire $f(x_1; x_2) = x_1^2 + x_2^2$

$$DL = f(\underline{x_0}) + \frac{\partial f}{\partial x_1}(\underline{x_0})(x_1 - x_{0,1}) + \frac{\partial f}{\partial x_2}(\underline{x_0})(x_2 - x_{0,2}) + \dots$$



$\vec{\nabla} f(\underline{x_0})$ projection du vecteur $\vec{N}(M_0)$ dans le plan des variables } gradient

↳ " s'éloigne du minimum en quelque sorte " (cas quadratique ici)



Exemple : $f(x) = \frac{1}{2} x_1^2 + 2x_2^2$

Le gradient vaut $\nabla f = \begin{pmatrix} x_1 \\ 4x_2 \end{pmatrix}$; $d_1 = \nabla f \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$

et $d_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ et $d_3 = \begin{pmatrix} -1 \\ 3 \end{pmatrix}$

Donc $d_1^t . \nabla f \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 17$; $d_2^t . \nabla f \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 5$ et $d_3^t . \nabla f \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 11$

1.4 Vitesse de convergence d'un algorithme

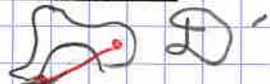
Exemple : La convergence superlinéaire d'ordre $p = 2$

- Supposons que l'erreur est $\varepsilon_k = 10^{-3}$ et $M = 10$
- A l'itération suivante $\varepsilon_{k+1} = 10^{-5}$ (l'erreur est divisée par 100 en 1 itération)
- Une nouvelle itération $\varepsilon_{k+2} = 10^{-9}$ (diapo 13)

1.6. Convexité : (diapo 16)

Ensemble convexe : L'ensemble des points appartient au domaine

Ensemble non convexe : L'ensemble des points n'appartiennent pas à tout le domaine



Fonction définie positive : $x^T \cdot \nabla^2 f(x) \cdot x > 0$

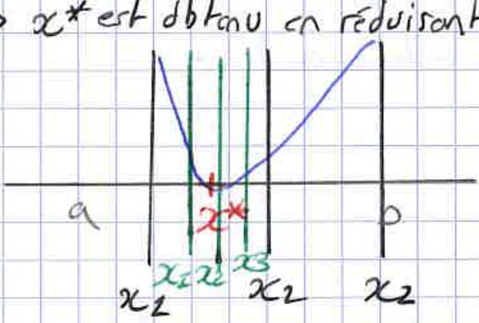
Fonction semi-définie positive : $x^T \cdot \Delta^2 f(x) \geq 0$

Fonction convexe :

1.7. Optimisation d'une fonction d'une variable réelle (diapo 17)

Fonction Unimodale :

Remarque : Trouver le minimum peut alors se résoudre par dichotomie (mais danger)
↳ x^* est obtenu en réduisant progressivement l'intervalle de recherche.



Exercice : Pourrez-vous sauvez le monde ? (diapo 22)

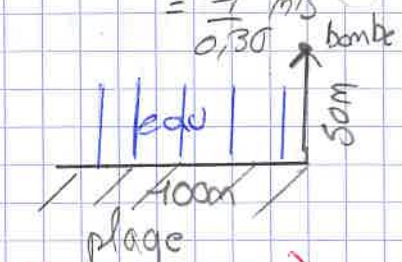
Données : $v_c = 18$ km/h $= 5$ m/s

$v_N = 10$ km/h $= \frac{1}{0,36}$ m/s

Les variables de décisions : d_c et d_N distances plage / x et distance nage

$t = v_c \cdot d_c + v_N \cdot d_N$ $\boxed{v = \frac{d}{t}}$!

est le temps pour arriver à la bome



On cherche à minimiser $\min_{(d_c;d_N)} v_c \cdot d_c + v_N \cdot d_N$

Les contraintes sont $0 \leq x \leq 100$ m

On ne minimise que le temps de course sur la plage

fonction $t_{min} = \min_{(x)} \left\{ \frac{x}{5} + 0,36 \cdot [(100 - x)^2 + (50)^2] \right\}$

Solution : $t_{min} = 34,96$ s

$x \simeq 66,59$ m