

Techniques d'optimisation

Max CERF

Techniques d'optimisation

Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
3. Optimisation avec contraintes

Sommaire

- 1. Bases théoriques**
 - 1.1 Définitions
 - 1.2 Contraintes linéaires
 - 1.3 Contraintes non linéaires
 - 1.4 Conditions d'optimalité
2. Optimisation sans contraintes
3. Optimisation avec contraintes

Sommaire

- 1. Bases théoriques**
 - 1.1 Définitions**
 - 1.1.1 Problème d'optimisation
 - 1.1.2 Solution
 - 1.1.3 Différentiabilité
 - 1.1.4 Convexité
 - 1.1.5 Conditionnement
 - 1.1.6 Direction de déplacement
 - 1.2 Contraintes linéaires
 - 1.3 Contraintes non linéaires
 - 1.4 Conditions d'optimalité
2. Optimisation sans contraintes
3. Optimisation avec contraintes

1.1.1 Problème d'optimisation

- ❑ Classification des problèmes d'optimisation
 - Optimisation continue dans \mathbb{R}^n
- ❑ Formulation mathématique et notations
 - Variables
 - Critère
 - Contraintes
- ❑ Norme sur \mathbb{R}^n
 - Norme vectorielle
 - Norme matricielle
- ❑ Suite dans \mathbb{R}^n
 - Limite
 - Vitesse de convergence

1.1.1 Classification

Optimisation fonctionnelle / paramétrique

- Inconnues = fonctions → Optimisation fonctionnelle
Optimisation en dimension infinie
Commande optimale
- Inconnues = entiers ou réels → Optimisation paramétrique
Optimisation en dimension finie
Programmation mathématique

Programmation mathématique

- Inconnues = entiers → Optimisation combinatoire
Programmation en nombres entiers
- Inconnues = réels → **Optimisation continue**
Programmation linéaire (LP)
Programmation non linéaire (NLP)
- Inconnues = entiers et réels → Programmation mixte

1.1.1 Formulation

Formulation mathématique

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sous} \quad \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \\ x \in X \end{cases}$$

→ formulation standard
problème noté (PO)

Notations

- x : n **variables** ou paramètres ou inconnues → vecteur de \mathbb{R}^n
- f : **critère** ou fonction **coût** ou fonction objectif → fonction de \mathbb{R}^n dans \mathbb{R}
 $x \in \mathbb{R}^n \mapsto f(x) \in \mathbb{R}$
- c_E : p **contraintes d'égalité** → fonction de \mathbb{R}^n dans \mathbb{R}^p
 $x \in \mathbb{R}^n \mapsto c_E(x) \in \mathbb{R}^p$
- c_I : q **contraintes d'inégalité** → fonction de \mathbb{R}^n dans \mathbb{R}^q
 $x \in \mathbb{R}^n \mapsto c_I(x) \in \mathbb{R}^q$
- X : ensemble convexe $X \subseteq \mathbb{R}^n$ → valeurs admissibles des variables

1.1.1 Optimisation continue

Hypothèses

- Continuité : Fonctions continues de variables réelles
→ Optimisation continue
≠ Optimisation combinatoire, Programmation en nombres entiers
- Différentiabilité : Fonctions différentiables
→ **Méthodes à base de gradient**
≠ Méthodes sans dérivées
- Déterminisme : Les données du problème sont parfaitement connues
≠ Optimisation stochastique
- Programmation linéaire : coût linéaire et contraintes linéaires (LP)
- Programmation quadratique : coût quadratique et contraintes linéaires (QP)
- Programmation non linéaire : cas général, fonctions quelconques (**NLP**)

Rappels d'analyse

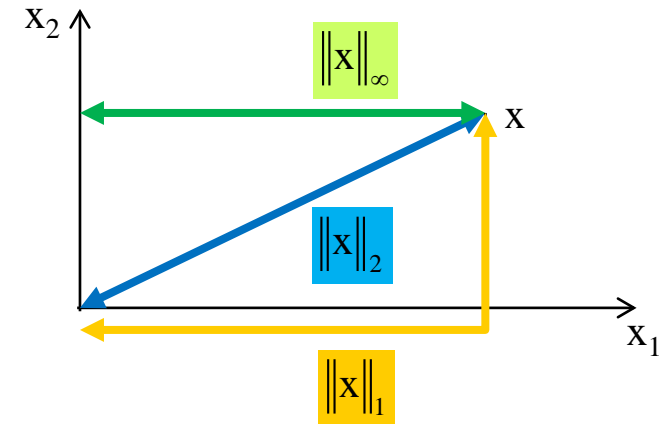
- Norme
- Suite - Convergence

1.1.1 Norme

Norme vectorielle sur \mathbb{R}^n

- Fonction $\| \cdot \| : \mathbb{R}^n \rightarrow \mathbb{R}$ vérifiant

$$\begin{cases} \|x\| \geq 0 \\ \|x\| = 0 \Leftrightarrow x = 0 \\ \|x + y\| \leq \|x\| + \|y\| \\ \|\alpha x\| = |\alpha| \|x\| \end{cases}$$
- Norme p : $\|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$
- Norme ∞** : $\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$
- Norme 2** = norme euclidienne



Norme matricielle

- Norme induite sur $\mathbb{R}^{m \times n}$ par la norme vectorielle $\| \cdot \|$
- Fonction $\| \cdot \|_{m \times n} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ définie par $\|A\|_{m \times n} = \max_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|}{\|x\|}$

1.1.1 Suite

Suite dans \mathbf{R}^n

- Suite : $\{x_k, k=0,1,2,\dots\} = \{x_0, x_1, x_2, \dots, x_n, \dots\}$
- Limite : $\lim_{k \rightarrow \infty} x_k = x^* \Leftrightarrow \lim_{k \rightarrow \infty} \|x_k - x^*\|$

Vitesse de convergence

- Convergence linéaire : $\|x_{k+1} - x^*\| \leq c \|x_k - x^*\|$ avec $0 \leq c < 1$
→ lent à partir d'un certain rang k_0
- Convergence superlinéaire : $\|x_{k+1} - x^*\| \leq c_k \|x_k - x^*\|$ avec $\lim_{k \rightarrow \infty} c_k = 0$
→ bonne à partir d'un certain rang k_0
- Convergence d'ordre p : $\|x_{k+1} - x^*\| \leq c \|x_k - x^*\|^p$ avec $0 \leq c < 1$
à partir d'un certain rang k_0
- **Convergence quadratique** si $p=2$
→ rapide

1.1.2 Solution

- ☐ Solution admissible
- ☐ Minimum
 - Minimum local
 - Minimum global
 - Infimum
- ☐ Problèmes équivalents
 - Transformations simples
- ☐ Contrainte active
- ☐ Point intérieur

1.1.2 Solution admissible

Solution admissible

x solution admissible de (PO) $\Leftrightarrow x$ satisfait les contraintes
 (ou point admissible)

$$\begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \\ x \in X \end{cases}$$

Ensemble admissible

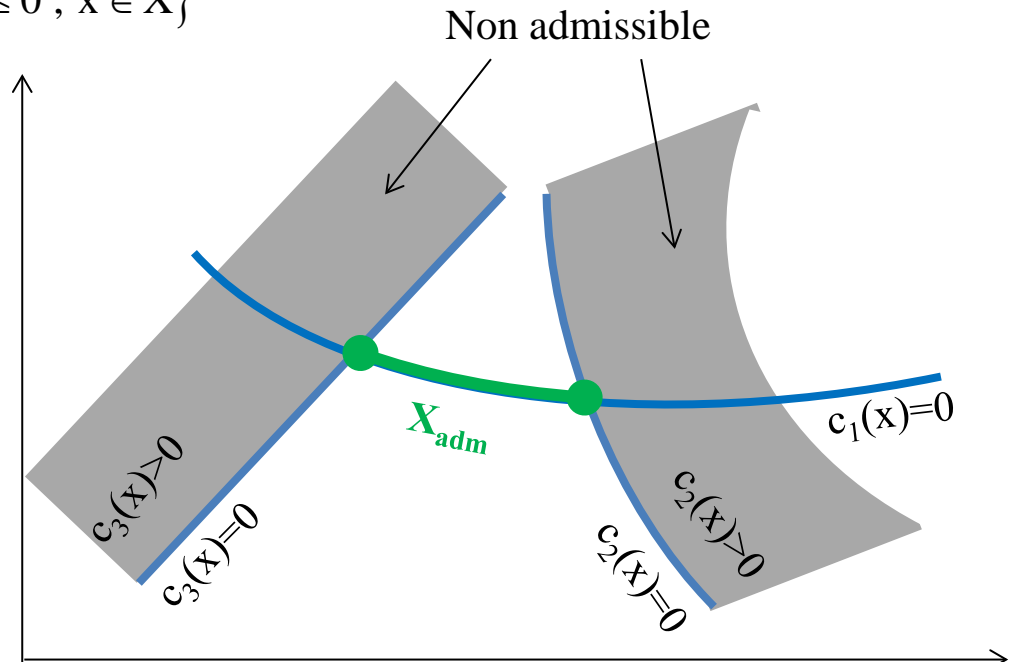
$$X_{\text{adm}} = \{x \in \mathbb{R}^n / c_E(x) = 0, c_I(x) \leq 0, x \in X\}$$

Illustration dans \mathbb{R}^2

- $c_1(x) = 0 \rightarrow$ courbe
- $c_2(x) \leq 0 \rightarrow$ région du plan
- $c_3(x) \leq 0 \rightarrow$ région du plan

Dans \mathbb{R}^n

- $c(x) = 0 \rightarrow$ **hypersurface**
(dimension $n-1$)
- $a^T x = 0 \rightarrow$ **hyperplan** $\perp a \in \mathbb{R}^n$
(linéaire)



1.1.2 Minimum

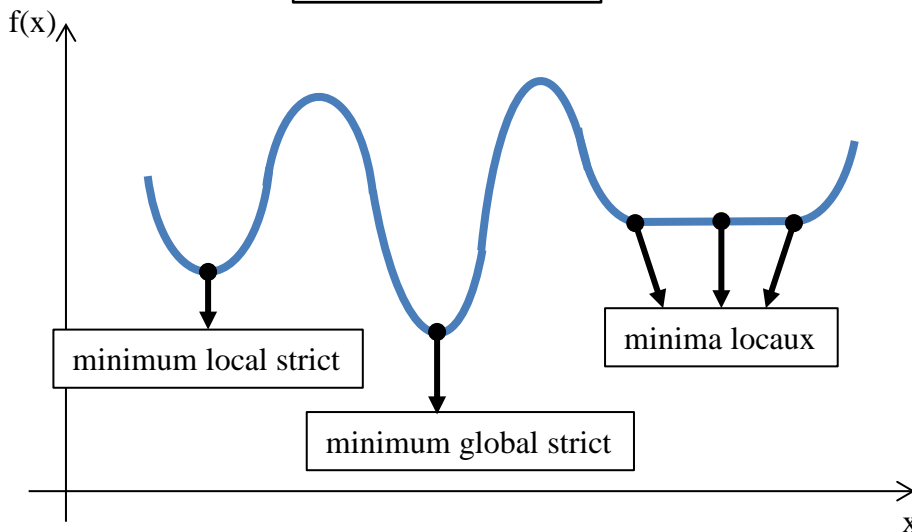
Minimum global (= meilleure solution dans l'absolu)

- x^* minimum global de (PO) $\Leftrightarrow \forall x \in X_{\text{adm}}, f(x^*) \leq f(x)$
- x^* minimum global strict $\Leftrightarrow \forall x \in X_{\text{adm}}, f(x^*) < f(x)$ si $x \neq x^*$

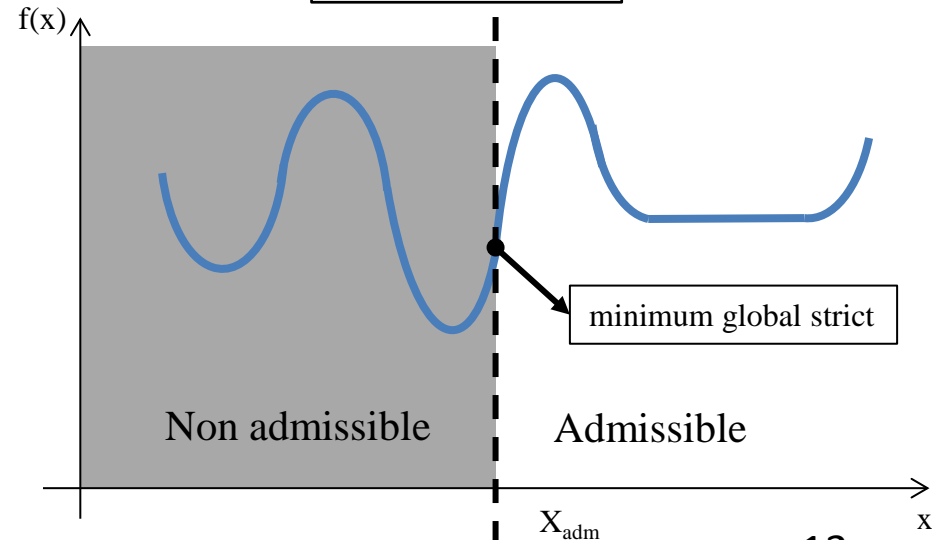
Minimum local (= meilleure solution dans un voisinage)

- x^* minimum local de (PO) $\Leftrightarrow \exists \varepsilon > 0 / \forall x \in X_{\text{adm}}, \|x - x^*\| < \varepsilon, f(x^*) \leq f(x)$
- x^* minimum local strict $\Leftrightarrow \exists \varepsilon > 0 / \forall x \in X_{\text{adm}}, \|x - x^*\| < \varepsilon, f(x^*) < f(x)$ si $x \neq x^*$

Sans contrainte



Avec contrainte



1.1.2 Infimum

Borne inférieure

$$f : \mathbb{R}^n \rightarrow \mathbb{R} \quad f \text{ bornée inférieurement sur } Y \subseteq \mathbb{R}^n \\ \Leftrightarrow \exists M \in \mathbb{R} / \forall x \in Y, M \leq f(x)$$

Infimum

- Infimum de f sur Y = plus grande borne inférieure
- Notation : $\inf_Y f = \inf \{f(y), y \in Y\}$
- Propriété :
$$\begin{cases} \forall x \in Y, \inf_Y f \leq f(x) \\ \text{et} \\ \forall M > \inf_Y f, \exists x \in Y / f(x) < M \end{cases}$$

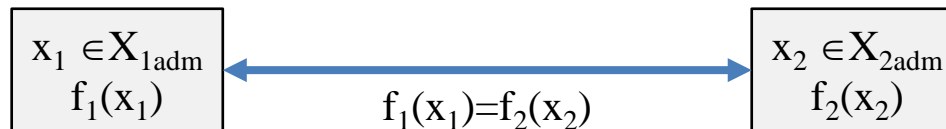
Théorème de Weierstrass

- f atteint son infimum si f continue, Y compact : $\exists x^* \in Y / f(x^*) = \inf_Y f$
- Conditions réalisées en pratique : fonctions continues, intervalles fermés
→ Le problème (PO) admet une solution x^* .

1.1.2 Problèmes équivalents

Problèmes équivalents

(PO_1) et (PO_2) sont deux problèmes équivalents si on peut associer à tout point admissible x_1 de (PO_1) un point admissible x_2 de (PO_2) avec la même valeur pour le critère.



(PO_1) et (PO_2) ont alors des solutions de même coût : $f_1(x_1^*) = f_2(x_2^*)$

Transformations simples

- Changement de variable : $y = \varphi(x)$ avec φ strictement croissante sur X
- Maximisation / minimisation : $\max_x f(x) \Leftrightarrow \min_x (-f(x))$
- Contrainte inférieur / supérieur : $c(x) \geq 0 \Leftrightarrow -c(x) \leq 0$
- Variables positives : $x = x^+ - x^-$ avec $\begin{cases} x^+ \geq 0 \\ x^- \geq 0 \end{cases}$
- Variables d'écart : $c(x) \leq 0 \Leftrightarrow \begin{cases} c(x) + y = 0 \\ y \geq 0 \end{cases} \Leftrightarrow c(x) + z^2 = 0$

1.1.2 Contrainte active

Contrainte active

Une contrainte du problème (PO) est **active** (ou **saturée**) en x si elle s'annule en x .

Ensemble des contraintes actives

$$C_{\text{act}}(x) = \{j / c_{Ej}(x) = 0\} \cup \{j / c_{Ij}(x) = 0\}$$

- Contrainte égalité c_E : x admissible $\Rightarrow c_E(x) = 0 \Rightarrow c_E$ active en x
- Contrainte inégalité c_I : x admissible $\Rightarrow c_I(x) \leq 0 \Rightarrow c_I$ active en x si $c_I(x) = 0$
 c_I inactive en x si $c_I(x) < 0$

Intérêt

- Les contraintes inégalité inactives n'ont pas d'influence sur la solution x^* du problème (PO). On peut les ignorer, si on identifie l'ensemble $C_{\text{act}}(x^*)$. Mais x^* n'est pas connu au départ ...
- **Le problème (PO) est équivalent au problème $(PO)_{\text{act}}$ réduit aux contraintes actives prises comme des contraintes égalité.**

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c_j(x) = 0, j \in C_{\text{act}}(x^*) \quad \text{noté} \quad \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

1.1.2 Exemple

Contrainte active

$$\min_{x \in \mathbb{R}} x^2 + 1 \text{ sous } \begin{cases} x \geq 1 \\ x \leq 2 \end{cases} \rightarrow x^* = 1$$

1. Minimum sans contrainte

$$\min_{x \in \mathbb{R}} x^2 + 1 \rightarrow x^* = 0$$

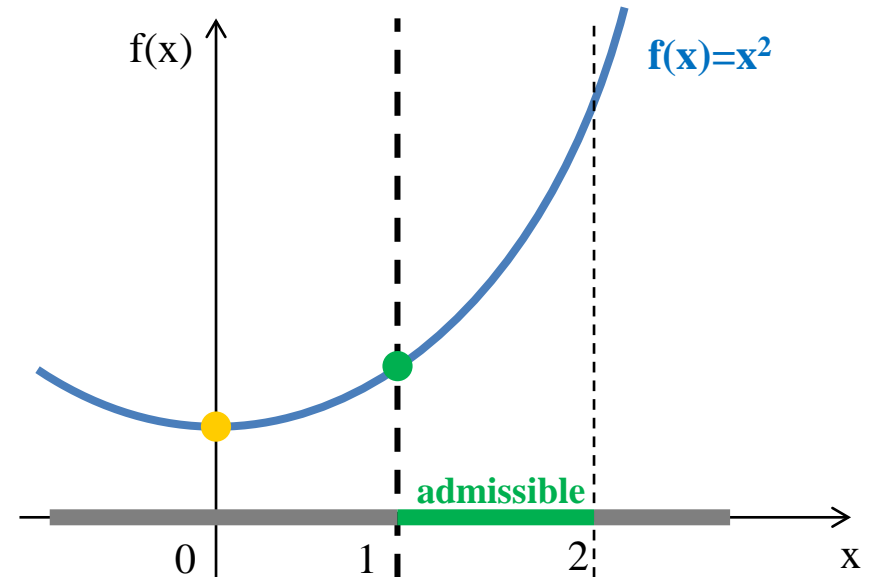
- Respecte la contrainte $x \leq 2$
- Ne respecte pas la contrainte $x \geq 1$
 → Activation de la contrainte $x = 1$

2. Minimum avec contrainte active $x = 1$

$$\min_{x \in \mathbb{R}} x^2 + 1 \text{ sous } x = 1 \rightarrow x^* = 1$$

- Respecte la contrainte $x \leq 2$
- Respecte la contrainte $x \geq 1$
 → Solution du problème

3. Bilan : 1 contrainte active $x \geq 1$ → transformée en égalité
 1 contrainte inactive $x \leq 2$ → ignorée



● Minimum sans contrainte

● Minimum avec contrainte

1.1.2 Point intérieur

Point intérieur

y point intérieur à Y

\Leftrightarrow Il existe un voisinage de y contenu dans Y : $\exists \varepsilon > 0 / \forall z, \|z - y\| \leq \varepsilon, z \in Y$

Un problème avec contraintes égalité n'admet pas de point intérieur

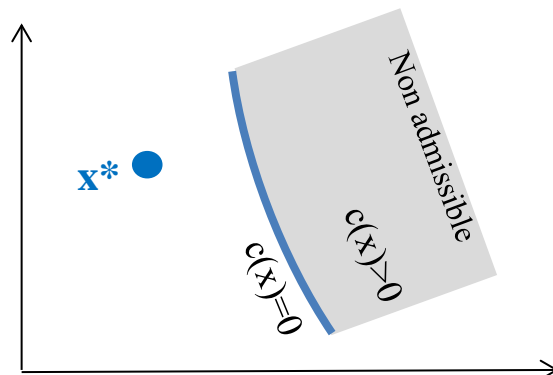
Solution intérieure aux contraintes

- x^* minimum local du problème avec contraintes inégalité
- Si x^* est un point intérieur, alors
 x^* minimum local du problème sans contraintes

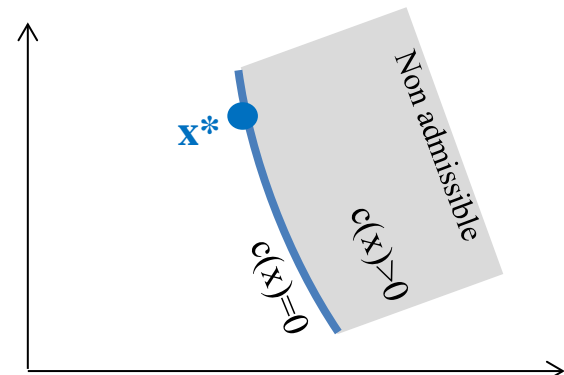
$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c_1(x) \leq 0$$

$$\min_{x \in \mathbb{R}^n} f(x) \rightarrow \text{plus simple}$$

Point intérieur - Contrainte inactive



Contrainte active



1.1.2 Exemple

Point intérieur

$$\min_{x \in \mathbb{R}} x^2 + 1 \text{ sous } x \leq 1 \rightarrow x^* = 0$$

1. Ensemble admissible

$$X_{\text{adm}} = \{x \in \mathbb{R} / x \leq 1\} =]-\infty, 1]$$

2. Ensemble intérieur à la contrainte

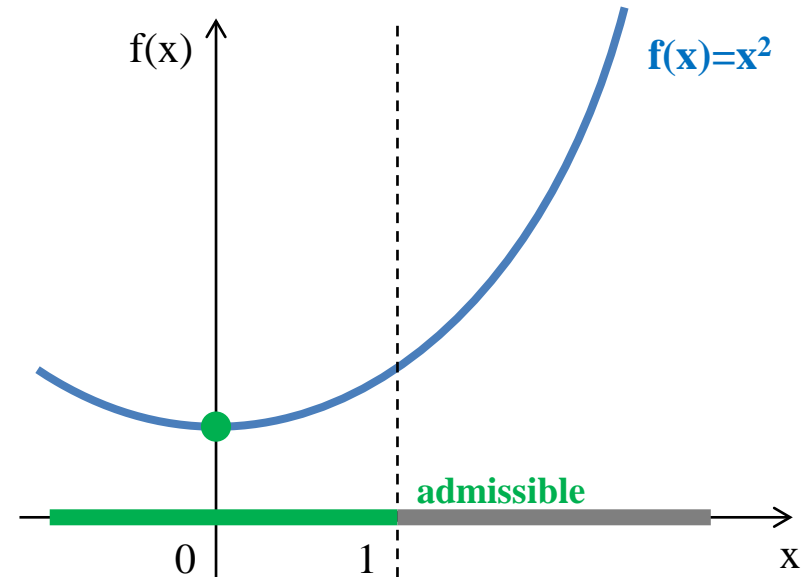
$$X_{\text{int}} = \{x \in \mathbb{R} / x < 1\} =]-\infty, 1[$$

$$X_{\text{int}} = X_{\text{adm}} - \{1\}$$

$x \in X_{\text{int}} \rightarrow$ voisinage de x inclus dans X_{int}
 \rightarrow intervalle ouvert

3. Solution : $x^* = 0$

$x^* \in X_{\text{int}}$ intérieur à la contrainte \rightarrow contrainte inactive



● Minimum avec contrainte

1.1.3 Différentiabilité

- ❑ Définitions
 - Dérivée partielle
 - Dérivée directionnelle
 - Gradient
 - Hessien
 - Jacobien

- ❑ Théorème de Taylor
 - Ordre 1 et 2
 - Modèle quadratique-linéaire
 - Ligne de niveau

- ❑ Dérivées numériques
 - Différences finies
 - Erreurs numériques
 - Incrément

1.1.3 Gradient

Différentiabilité ordre 1

f fonction continue de \mathbb{R}^n dans \mathbb{R}

Dérivée partielle

Dérivée partielle de f en x par rapport à x_i : $f_{x_i}(x) = \frac{\partial f(x)}{\partial x_i} = \lim_{s \rightarrow 0} \frac{f(x_1, \dots, x_i + s, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{s}$
 si la limite existe

Gradient

Gradient de f en x : $g(x) = \nabla f(x)$

$$g(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

si toutes les dérivées partielles existent

$$g(x) = \left(\frac{\partial f(x)}{\partial x_i} \right)_{i=1, \dots, n} = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

Dérivée directionnelle

Dérivée directionnelle de f en x dans la direction $d \in \mathbb{R}^n$:

si la limite existe

(dérivée directionnelle = produit scalaire avec le gradient) \Rightarrow

$$f_d(x) = \lim_{s \rightarrow 0} \frac{f(x + sd) - f(x)}{s}$$

$$f_d(x) = g(x)^T d$$

Fonction différentiable

f différentiable en $x \Leftrightarrow f$ admet une dérivée directionnelle pour tout $d \in \mathbb{R}^n$

1.1.3 Hessien

Différentiabilité ordre 2

f fonction deux fois différentiable de \mathbb{R}^n dans \mathbb{R}

Hessien

Hessien de f en x : $H(x) = \nabla^2 f(x)$
 $H(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$

$$H(x) = \left(\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right)_{i,j=1,\dots,n} = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial^2 x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial^2 x_n} \end{pmatrix}$$

Courbure

On définit pour une direction $d \in \mathbb{R}^n$ au point x la fonction φ à une variable : $\varphi(s) = f(x+sd)$
 → variation de f dans la direction d

- Dérivée première : $\varphi'(s) = d^T \nabla f(x+sd) \rightarrow \varphi'(0) = d^T g(x)$
- Dérivée seconde : $\varphi''(s) = d^T \nabla^2 f(x+sd) d \rightarrow \varphi''(0) = d^T H(x) d$

La **courbure de f en x dans la direction d** est définie par : $\frac{d^T H(x) d}{d^T d}$
 → normalisation de φ'' en $s=0$

= quotient de Rayleigh de $H(x)$ dans la direction d

1.1.3 Jacobien

Matrice gradient

c fonction continue de \mathbb{R}^n dans \mathbb{R}^m

Gradient de c en x : $\nabla c(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$

$$\nabla c(x) = (\nabla c_1(x), \dots, \nabla c_m(x)) = \left(\frac{\partial c_j(x)}{\partial x_i} \right)_{\substack{i=1, \dots, n \\ j=1, \dots, m}} = \begin{pmatrix} \frac{\partial c_1(x)}{\partial x_1} & \dots & \frac{\partial c_m(x)}{\partial x_1} \\ \dots & \dots & \dots \\ \frac{\partial c_1(x)}{\partial x_n} & \dots & \frac{\partial c_m(x)}{\partial x_n} \end{pmatrix}$$

Matrice jacobienne (« jacobien » = déterminant de J_c)

$$J_c(x) = \nabla c(x)^T = \begin{pmatrix} \nabla c_1(x)^T \\ \dots \\ \nabla c_m(x)^T \end{pmatrix} = \left(\frac{\partial c_j(x)}{\partial x_i} \right)_{\substack{i=1, \dots, n \\ j=1, \dots, m}} = \begin{pmatrix} \frac{\partial c_1(x)}{\partial x_1} & \dots & \frac{\partial c_1(x)}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial c_m(x)}{\partial x_1} & \dots & \frac{\partial c_m(x)}{\partial x_n} \end{pmatrix}$$

Contraintes du problème (PO)

Matrice jacobienne regroupant les contraintes égalité c_E et les contraintes inégalité c_I

$$J(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{(p+q) \times n} \quad J(x) = \begin{pmatrix} J_E(x) \\ J_I(x) \end{pmatrix} = \begin{pmatrix} \nabla c_E(x)^T \\ \nabla c_I(x)^T \end{pmatrix}$$

1.1.3 Théorème de Taylor

Théorème de Taylor

f fonction de \mathbb{R}^n dans \mathbb{R} : $x \in \mathbb{R}^n \mapsto f(x) \in \mathbb{R}$
 $d \in \mathbb{R}^n$: déplacement à partir de x

- **Ordre 1**

f continument différentiable 1 fois au voisinage de x

$$f(x + d) = f(x) + \nabla f(x)^T d + o(\|d\|)$$

Il existe $s \in [0, 1]$ tel que : $f(x + d) = f(x) + \nabla f(x + sd)^T d$

- **Ordre 2**

f continument différentiable 2 fois au voisinage de x

$$f(x + d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d + o(\|d\|^2)$$

Il existe $s \in [0, 1]$ tel que : $f(x + d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x + sd) d$

1.1.3 Modèle quadratique-linéaire

Problème avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \rightarrow \text{contrainte actives}$$

Fonction modèle

- Application du théorème de Taylor au point $x_0 \in \mathbb{R}^n$
- **Modèle quadratique du critère** : $\hat{f}_0(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(x_0) (x - x_0)$
- **Modèle linéaire des contraintes** : $\hat{c}_0(x) = c(x_0) + \nabla c(x_0)^T (x - x_0)$

Problème quadratique-linéaire local

$$\text{Au voisinage de } x_0 : \begin{cases} f(x) \approx \hat{f}_0(x) \\ c(x) \approx \hat{c}_0(x) \end{cases}$$

$$\min_{x \in \mathbb{R}^n} \hat{f}_0(x) \text{ sous } \hat{c}_0(x) = 0 \rightarrow \text{Problème quadratique-linéaire}$$

- localement « proche » du problème initial
- plus simple à résoudre

1.1.3 Exemple

Modèle quadratique : fonction de 1 variable

- Fonction : $f(x) = -x^4 + 12x^3 - 47x^2 + 60x$
- Gradient : $\nabla f(x) = -4x^3 + 36x^2 - 94x + 60$
- Hessien : $\nabla^2 f(x) = -12x^2 + 72x - 94$
- Modèle quadratique en $x_0 = 3$: $f(x_0) = 0$, $\nabla f(x_0) = -6$, $\nabla^2 f(x_0) = 14$
$$\hat{f}_0(x) = -6(x-3) + \frac{1}{2}14(x-3)^2 = 7x^2 - 48x + 81$$
- Modèle quadratique en $x_0 = 4$: $f(x_0) = 0$, $\nabla f(x_0) = 4$, $\nabla^2 f(x_0) = 2$
$$\hat{f}_0(x) = 4(x-4) + \frac{1}{2}2(x-4)^2 = x^2 - 4x$$
- Modèle quadratique en $x_0 = 5$: $f(x_0) = 0$, $\nabla f(x_0) = -10$, $\nabla^2 f(x_0) = -34$
$$\hat{f}_0(x) = -10(x-5) - \frac{1}{2}34(x-5)^2 = -17x^2 + 160x - 375$$

1.1.3 Exemple

Modèle quadratique : fonction de 2 variables

- Fonction : $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \rightarrow$ **fonction de Rosenbrock**
- Gradient : $\nabla f(x) = \begin{pmatrix} -400(x_2 - x_1^2)x_1 - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix}$
- Hessien : $\nabla^2 f(x) = \begin{pmatrix} -400(x_2 - 3x_1) + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}$
- Modèle quadratique en $x_0 = (1, 1)$: $f(x_0) = 0$, $\nabla f(x_0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\nabla^2 f(x_0) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}$

$$\hat{f}_0(x_1, x_2) = \frac{1}{2} \begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix}^T \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix}$$

$$\Rightarrow \hat{f}_0(x_1, x_2) = 401(x_1 - 1)^2 - 400(x_1 - 1)(x_2 - 1) + 100(x_2 - 1)^2$$

1.1.3 Résultats utiles

Gradient d'une fonction scalaire

- Le gradient de $f : \mathbb{R}^n \rightarrow \mathbb{R}$ en x est le vecteur $g \in \mathbb{R}^n$ tel que

$$f(x+d) = f(x) + g^T d + o(\|d\|), \forall d \in \mathbb{R}^n$$

- Fonction quadratique** : $f(x) = \frac{1}{2} x^T Q x + c^T x + b$ Q matrice symétrique

$$f(x+d) = \frac{1}{2} (x+d)^T Q (x+d) + c^T (x+d) + b = f(x) + (Qx + c)^T d + \frac{1}{2} d^T Q d$$

$$\rightarrow \boxed{\nabla f(x) = Qx + c} \quad \boxed{\nabla^2 f(x) = Q}$$

Gradient d'une fonction vectorielle

- Le gradient de $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ en x est la matrice $G \in \mathbb{R}^{n \times p}$ telle que

$$f(x+d) = f(x) + G^T d + o(\|d\|), \forall d \in \mathbb{R}^n$$

- Fonction composée** : $f(x) = h(g(x))$ avec $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $h : \mathbb{R}^m \rightarrow \mathbb{R}^p$

$$f(x+d) = h(g(x+d)) = h(g(x) + \nabla g(x)^T d + o(\|d\|)) = h(g(x)) + \nabla h(g(x))^T \nabla g(x)^T d + o(\|d\|)$$

$$\rightarrow \boxed{\nabla f(x) = \nabla g(x) \nabla h(g(x))}$$

- Fonction linéaire** : $g(x) = Ax \Rightarrow g(x+d) = Ax + Ad \Rightarrow \nabla g(x) = A^T$

$$\boxed{f(x) = h(Ax) \rightarrow \nabla f(x) = A^T \nabla h(Ax)}$$

1.1.3 Ligne de niveau

Définition

- **Ligne (ou surface) de niveau l_0 de la fonction f**

$L_0 = \{x \in \mathbb{R}^n / f(x) = l_0\} \rightarrow$ hypersurface dans \mathbb{R}^n (sous-espace de dimension $n-1$)

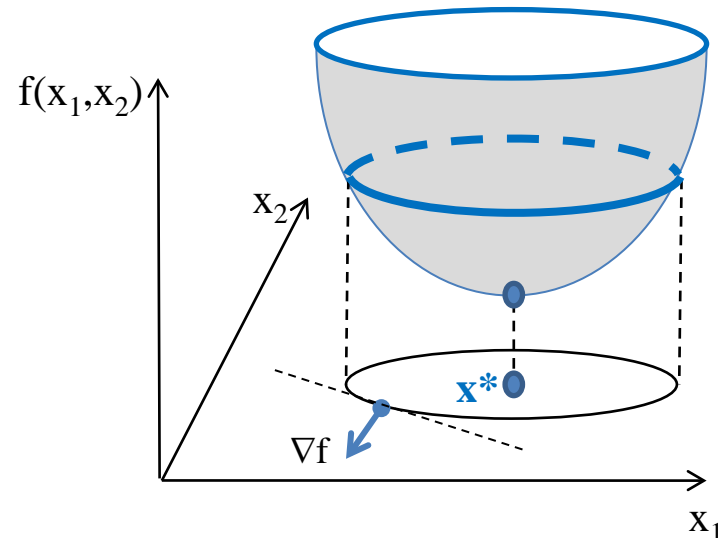
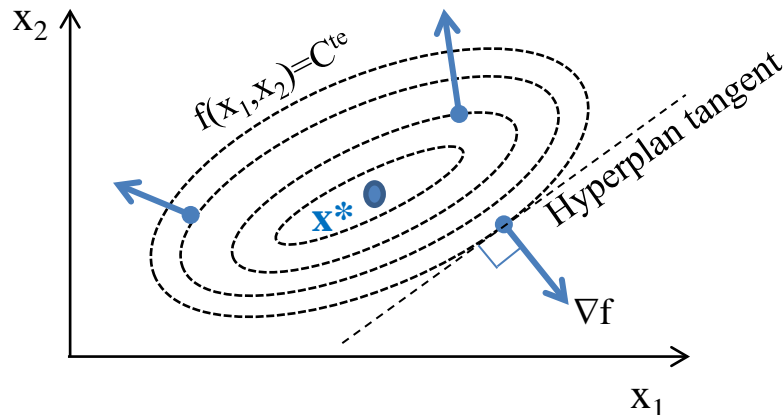
- Ligne de niveau passant par x_0

$L_0 = \{x \in \mathbb{R}^n / f(x) = f(x_0)\}$ avec $f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + o(\|x - x_0\|)$ à l'ordre 1

Gradient

$x \in L_0 \Rightarrow f(x) = f(x_0) \Rightarrow \nabla f(x_0)^T (x - x_0) = 0 \rightarrow$ hyperplan tangent à L_0 en x_0

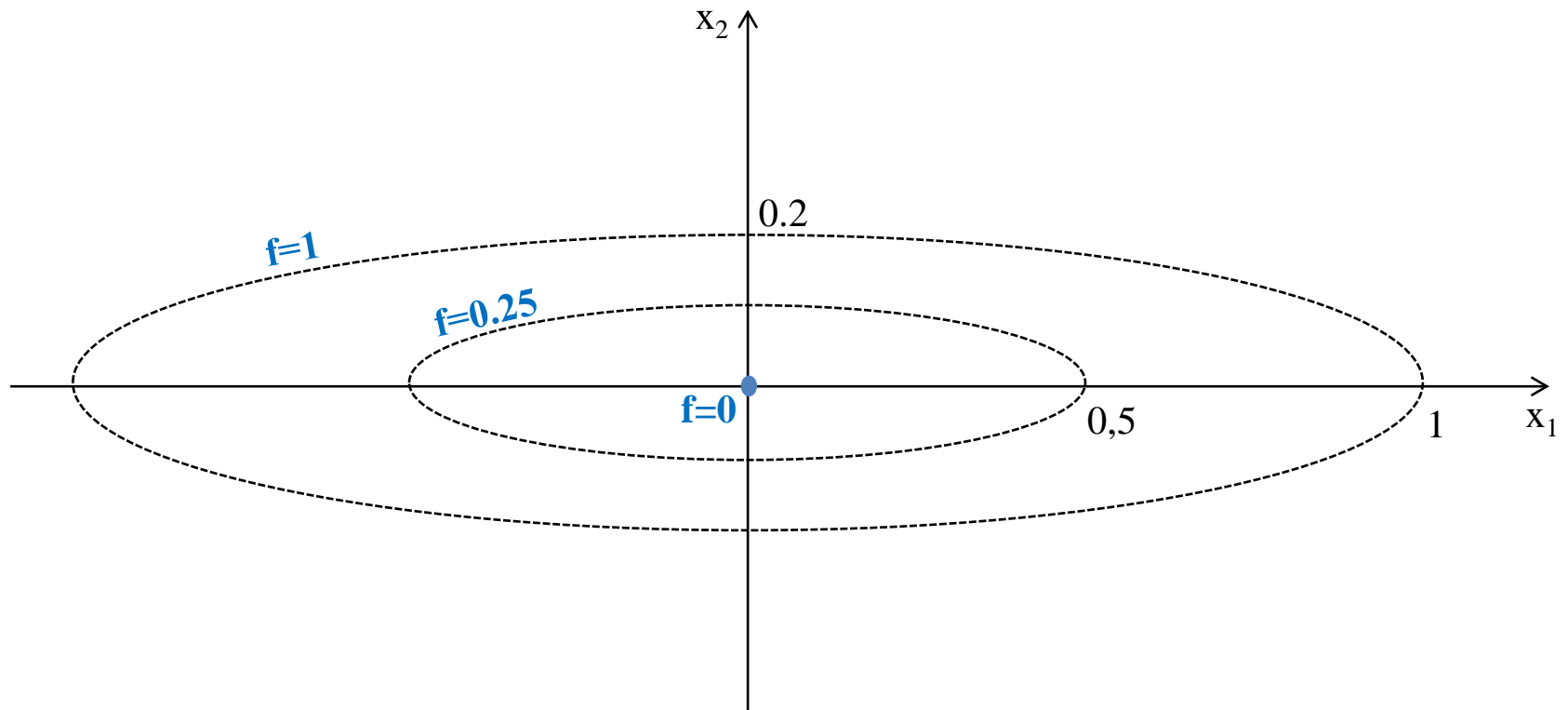
Le gradient de f est orthogonal aux lignes de niveaux.



1.1.3 Exemple

Ligne de niveau : fonction quadratique

$$f(x_1, x_2) = x_1^2 + 25x_2^2$$

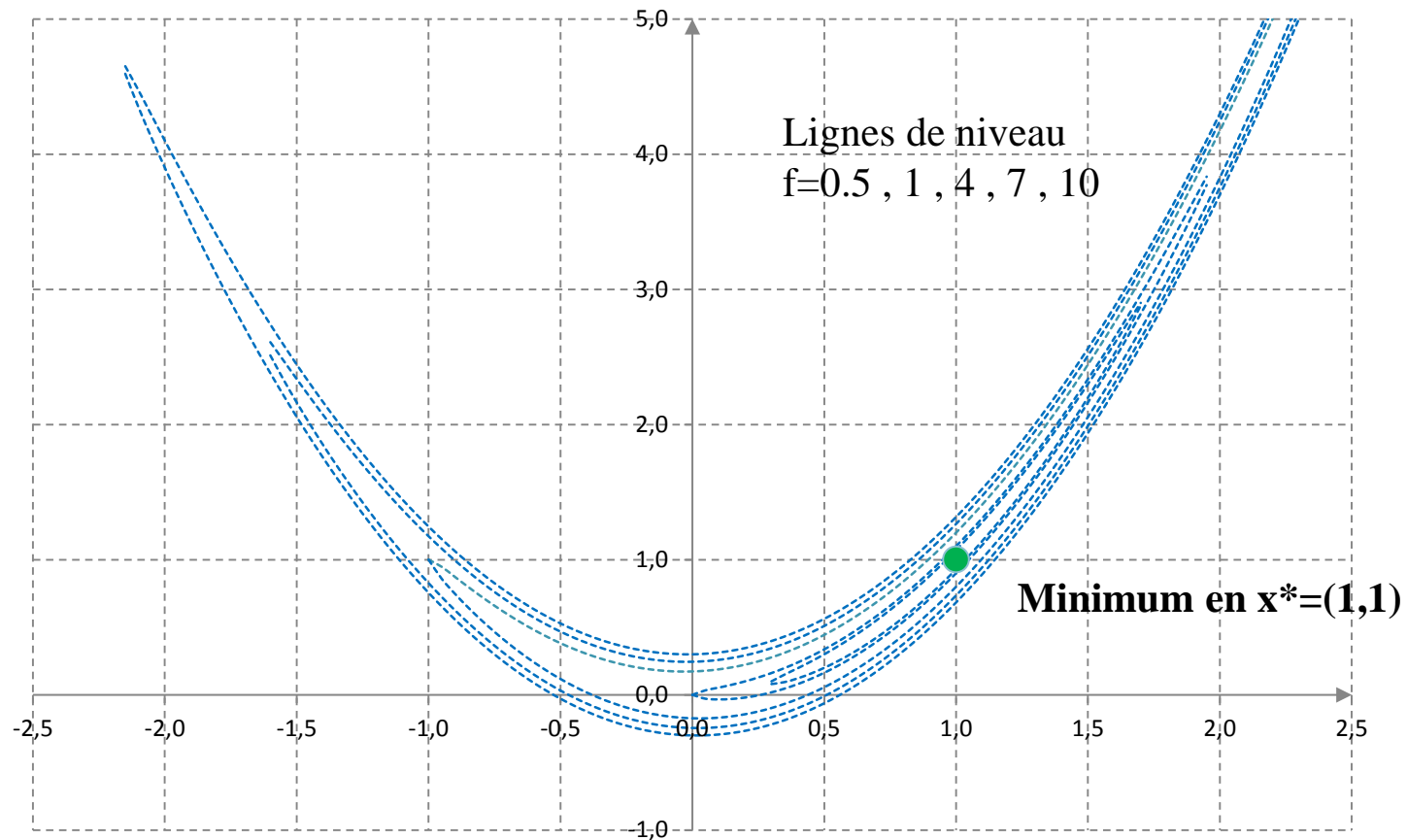


Techniques d'optimisation

1.1.3 Exemple

Ligne de niveau : fonction de Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$



1.1.3 Dérivées numériques

Différences finies

Les expressions analytiques de $\nabla f(x)$ et $\nabla^2 f(x)$ ne sont généralement pas disponibles.

→ Evaluation par différences finies avec incrément h appliqué successivement sur chaque variable

$$x \rightarrow x + he_i \quad \text{avec} \quad e_i = (0, \dots, 0, 1, 0, \dots, 0)^T, \quad i = 1 \text{ à } n$$

Gradient

- Différence finie simple : $\frac{\partial f}{\partial x_i}(x) = \frac{f(x + he_i) - f(x)}{h} + o(h)$
 amont si $h < 0$
 aval si $h > 0$
 → n appels fonction pour évaluer $\nabla f(x)$
- Différence finie centrée : $\frac{\partial f}{\partial x_i}(x) = \frac{f(x + he_i) - f(x - he_i)}{2h} + o(h^2)$
 plus précis
 → $2n$ appels fonction pour évaluer $\nabla f(x)$

Hessien

- Différence finie simple : $\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \frac{f(x + he_i + he_j) - f(x + he_i) - f(x + he_j) + f(x)}{h^2} + o(h)$
 → $n(n+1)/2 + n$ appels fonction pour évaluer $\nabla^2 f(x)$

1.1.3 Dérivées numériques

Sources d'erreurs

L'évaluation d'une dérivée par différence finie génère 2 types d'erreurs :

- Erreur d'arrondi (ou de conditionnement)
- Erreur de troncature

Erreur d'arrondi

Les réels sont représentés en mémoire machine calcul avec une précision finie.

La précision machine ε_m est le plus petit réel tel que : $1 + \varepsilon_m \neq 1$

→ erreur relative $\varepsilon_m = 10^{-16}$ sur la valeur d'un réel x en double précision

→ erreur relative ε_r sur la valeur évaluée de $f(x)$

$\varepsilon_r \gg \varepsilon_m$ par cumul des erreurs au cours des opérations pour passer de x à $f(x)$

$f_{\text{eval}}(x) = f_{\text{exact}}(x)(1 \pm \varepsilon_r) = f_{\text{exact}}(x) \pm \varepsilon_f \rightarrow \varepsilon_f = \text{erreur absolue sur } f$

Erreur de troncature

L'évaluation d'une dérivée par différence finie tronque le développement de Taylor à l'ordre 1.

- $f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{1}{2}h^2f''(x_0 + sh)$ avec $s \in [0,1]$
- $f(x_0 + h) = f(x_0) + hf'(x_0) \pm \frac{1}{2}h^2\varepsilon_t$ avec $\varepsilon_t < M$ majorant de $|f''(x)|$ sur $[x_0, x_0 + d]$

1.1.3 Dérivées numériques

Erreur sur la dérivée

$$f'_{\text{eval}}(x_0) = \frac{f_{\text{eval}}(x_0 + h) - f_{\text{eval}}(x_0)}{h} \quad \text{avec}$$

- $f_{\text{eval}}(x_0) = f_{\text{exact}}(x_0) \pm \varepsilon_f \rightarrow$ arrondi sur $f_{\text{eval}}(x_0)$
- $f_{\text{eval}}(x_0 + h) = f_{\text{exact}}(x_0 + h) \pm \varepsilon_f \rightarrow$ arrondi sur $f_{\text{eval}}(x_0 + h)$
- $f_{\text{exact}}(x_0 + h) = f_{\text{exact}}(x_0) + hf'_{\text{exact}}(x_0) \pm \frac{1}{2}h^2\varepsilon_t \rightarrow$ troncature sur $f_{\text{exact}}(x_0 + h)$

En remplaçant dans l'expression de $f'_{\text{eval}}(x_0)$:

$$\Rightarrow f'_{\text{eval}}(x_0) = \frac{f_{\text{exact}}(x_0 + h) - f_{\text{exact}}(x_0) \pm 2\varepsilon_f}{h} = \frac{hf'_{\text{exact}}(x_0) \pm \frac{1}{2}h^2\varepsilon_t \pm 2\varepsilon_f}{h}$$

$$\Rightarrow f'_{\text{eval}}(x_0) = f'_{\text{exact}}(x_0) \pm \frac{h\varepsilon_t}{2} \pm \frac{2\varepsilon_f}{h}$$

L'erreur maximale sur la dérivée numérique est :

$$\varepsilon_{f'} = \frac{h\varepsilon_t}{2} + \frac{2\varepsilon_f}{h}$$

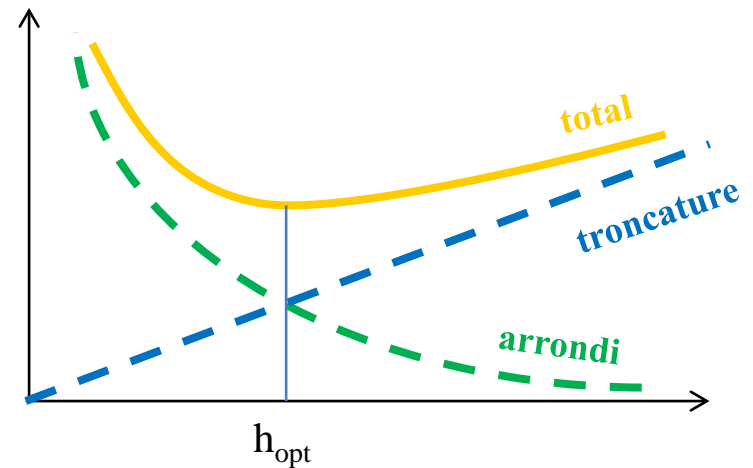
1.1.3 Dérivées numériques

Incrément optimal

- On choisit l'incrément pour minimiser l'erreur : $\min_h \varepsilon_{f'} = \frac{h\varepsilon_t}{2} + \frac{2\varepsilon_f}{h}$

$$\frac{d\varepsilon_{f'}}{dh} = \frac{\varepsilon_t}{2} - \frac{2\varepsilon_f}{h^2} = 0 \Rightarrow h_{\text{opt}} = 2\sqrt{\frac{\varepsilon_f}{\varepsilon_t}}$$

$$\Rightarrow \varepsilon_{f'} = 2\sqrt{\varepsilon_f \varepsilon_t}$$



- Règle empirique** de réglage de l'incrément
 En supposant que l'ordre de grandeur de la dérivée seconde est de l'ordre de 1 :

$$h_{\text{opt}} \approx \sqrt{\varepsilon_f}$$

→ incrément de l'ordre de la racine de la précision d'évaluation de f

$$\varepsilon_{f'} \approx \sqrt{\varepsilon_f}$$

→ précision sur f' de l'ordre de la racine de la précision d'évaluation de f
 (2 fois moins de chiffres significatifs)

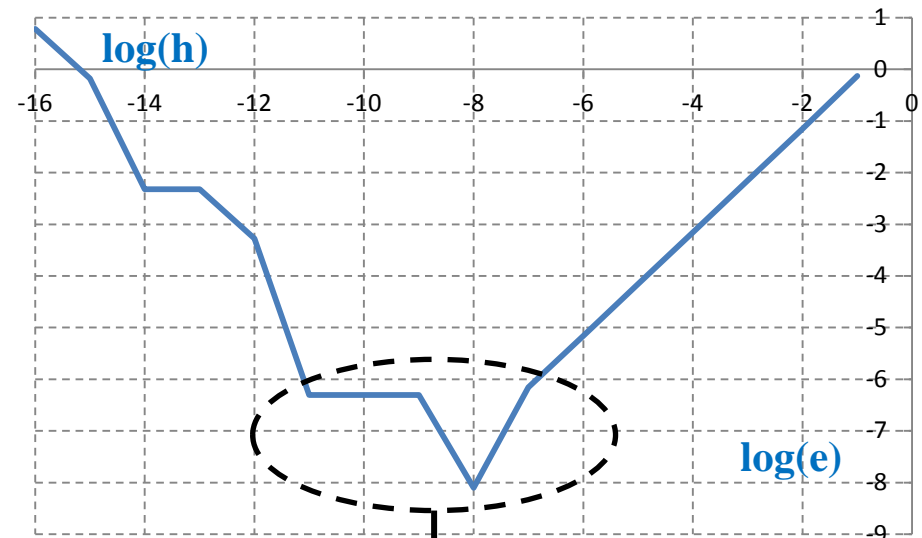
1.1.3 Exemple

Dérivée numérique

$$f(x) = x^4 + x^2 \rightarrow f'(x) = 4x^3 + 2x$$

- Dérivée en $x=1$: $f'(1) = 6$
- Dérivée numérique avec incrément $h \rightarrow$ erreur $e(h) = \left| \frac{f(x+h) - f(x)}{h} - f'(x) \right|$

h	$(f(x+h)-f(x))/h$	Erreur
1E-01	6,7410000000	7,410E-01
1E-02	6,0704010000	7,040E-02
1E-03	6,0070040010	7,004E-03
1E-04	6,0007000400	7,000E-04
1E-05	6,0000700004	7,000E-05
1E-06	6,0000069997	7,000E-06
1E-07	6,0000007007	7,007E-07
1E-08	6,0000000079	7,944E-09
1E-09	6,0000004964	4,964E-07
1E-10	6,0000004964	4,964E-07
1E-11	6,0000004964	4,964E-07
1E-12	6,0005334035	5,334E-04
1E-13	5,9952043330	-4,796E-03
1E-14	5,9952043330	-4,796E-03
1E-15	6,6613381478	6,613E-01



incrément adapté

1.1.4 Convexité

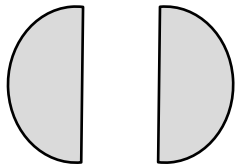
- ☐ Ensemble convexe
- ☐ Fonction convexe
- ☐ Lien avec le gradient et le hessien

1.1.4 Convexité

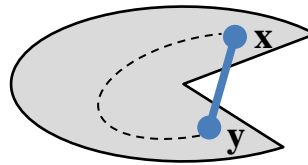
Ensemble convexe

$X \subseteq \mathbb{R}^n$ convexe $\Leftrightarrow \forall x, y \in X, \forall \lambda \in [0,1], \lambda x + (1-\lambda)y \in X$

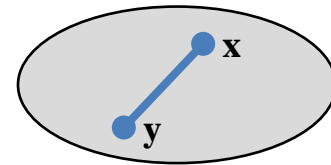
Interprétation géométrique : Segment inclus dans X



Non connexe



Non convexe



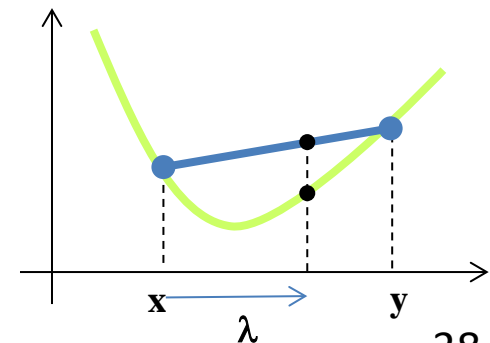
Convexe

Fonction convexe

f fonction de \mathbb{R}^n dans \mathbb{R}

- f convexe $\Leftrightarrow \forall x, y \in \mathbb{R}^n, \forall \lambda \in [0,1], f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$
- f strictement convexe $\Leftrightarrow \forall x, y \in \mathbb{R}^n, \forall \lambda \in]0,1[, f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda)f(y)$
- f concave $\Leftrightarrow -f$ convexe

Interprétation géométrique : Sécante au dessus de la courbe
Concavité vers le haut



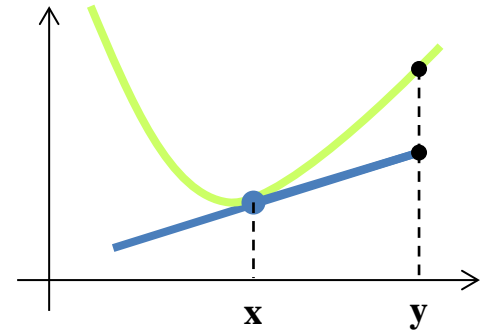
1.1.4 Convexité

Convexité et gradient

f fonction différentiable de $X \subseteq \mathbb{R}^n$ dans \mathbb{R} , X ensemble convexe ouvert

- f convexe $\Leftrightarrow \forall x, y \in X, f(y) - f(x) \geq (y - x)^T g(x)$
- f strictement convexe $\Leftrightarrow \forall x, y \in X, f(y) - f(x) > (y - x)^T g(x)$

Interprétation géométrique : Tangente au dessous de la courbe



Convexité et hessien

f fonction deux fois différentiable de $X \subseteq \mathbb{R}^n$ dans \mathbb{R} , X ensemble convexe ouvert

- f convexe $\Leftrightarrow \forall x \in \mathbb{R}^n, H(x)$ **semi-définie positive**
- f strictement convexe $\Leftrightarrow \forall x \in \mathbb{R}^n, H(x)$ définie positive

Matrice définie positive

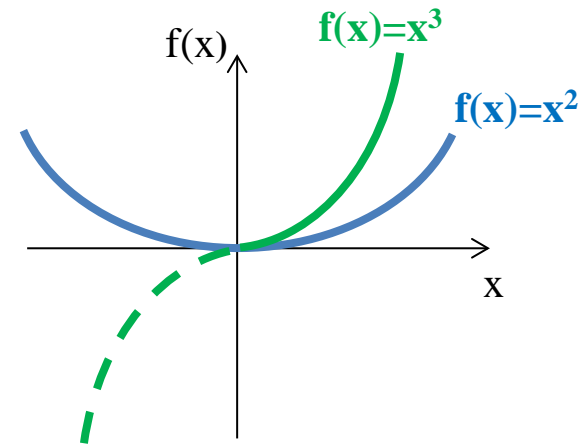
$A \in \mathbb{R}^{n \times n}$

- A définie positive $\Leftrightarrow \forall d \in \mathbb{R}^n, d^T A d > 0$
- A semi-définie positive $\Leftrightarrow \forall d \in \mathbb{R}^n, d^T A d \geq 0$

1.1.4 Exemple

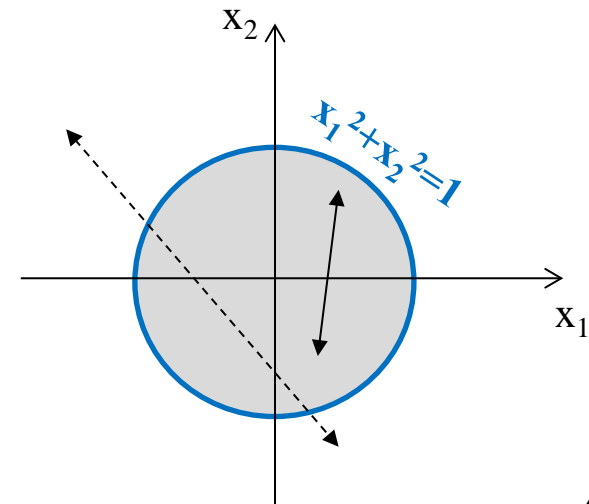
Fonction convexe

- Fonction : $f(x) = x^2$
 $f''(x) = 2 \rightarrow$ convexe sur \mathbb{R}
- Fonction : $f(x) = x^3$
 $f''(x) = 6x \rightarrow$ convexe sur \mathbb{R}^+
 \rightarrow non convexe sur \mathbb{R}



Ensemble convexe

- Ensemble : $X = \{(x_1, x_2) / x_1^2 + x_2^2 \leq 1\}$
 \rightarrow convexe
- Ensemble : $X = \{(x_1, x_2) / x_1^2 + x_2^2 \geq 1\}$
 \rightarrow non convexe



1.1.5 Conditionnement

- ☐ Conditionnement d'une matrice
- ☐ Conditionnement d'une fonction
- ☐ Préconditionnement
- ☐ Système linéaire perturbé
- ☐ Mise à l'échelle

1.1.5 Conditionnement

Conditionnement d'une matrice

A matrice symétrique semi-définie positive

Valeurs propres de A : $\sigma_1 \geq \dots \geq \sigma_n$ $\rightarrow \|A\|_2 = \sigma_1$

Nombre de conditionnement de A :

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_n} \geq 1$$

Conditionnement d'une fonction

f fonction deux fois différentiable

Conditionnement de f en x = nombre de conditionnement de $A=H(x)$

Interprétation

Vecteur propre d_k associé à la valeur propre σ_k : $A_k d_k = \sigma_k d_k$

Courbure de f en x dans la direction d_k : $\frac{d_k^T H(x) d_k}{d_k^T d_k} = \sigma_k$

Théorème de Rayleigh-Ritz

d_1 = direction de plus forte courbure (courbure = σ_1)

d_n = direction de plus faible courbure (courbure = σ_n)

1.1.5 Exemple

Matrice 2×2

- Inverse : $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$ avec $\det(A) = ad - bc$
- Valeurs propres : $\det(A - \sigma I) = 0 \Rightarrow (a - \sigma)(d - \sigma) - bc = 0 \Rightarrow \sigma^2 - (a + d)\sigma + \det(A) = 0$

Conditionnement

- $A = \begin{pmatrix} 0.1 & 1 \\ 0.20002 & 2 \end{pmatrix}$
- Valeurs propres : $\sigma^2 - 2.1\sigma - 0.00002 = 0 \Rightarrow \begin{cases} \sigma_1 = 2.10001 \\ \sigma_2 = -0.00001 \end{cases}$
- Conditionnement : $\begin{cases} \|A\|_2 = \sigma_1 \\ \|A^{-1}\|_2 = 1/\sigma_2 \end{cases} \Rightarrow \kappa(A) = \frac{\sigma_1}{\sigma_2} = 210001$

1.1.5 Préconditionnement

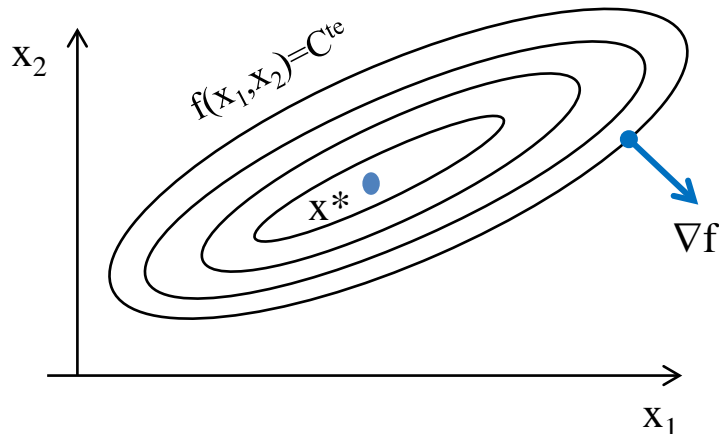
Changement de variable

- Variable : $\tilde{x} = Mx$ ($M \in \mathbb{R}^{n \times n}$ inversible = matrice de preconditionnement)
- Fonction : $\tilde{f}(\tilde{x}) = f(x) = f(M^{-1}\tilde{x})$
- Gradient : $\nabla \tilde{f}(\tilde{x}) = M^{-T} \nabla f(M^{-1}\tilde{x}) \Rightarrow \tilde{g}(\tilde{x}) = M^{-T} g(x)$
- Hessian : $\nabla^2 \tilde{f}(\tilde{x}) = M^{-T} \nabla^2 f(M^{-1}\tilde{x}) M^{-1} \Rightarrow \tilde{H}(\tilde{x}) = M^{-T} H(x) M^{-1}$

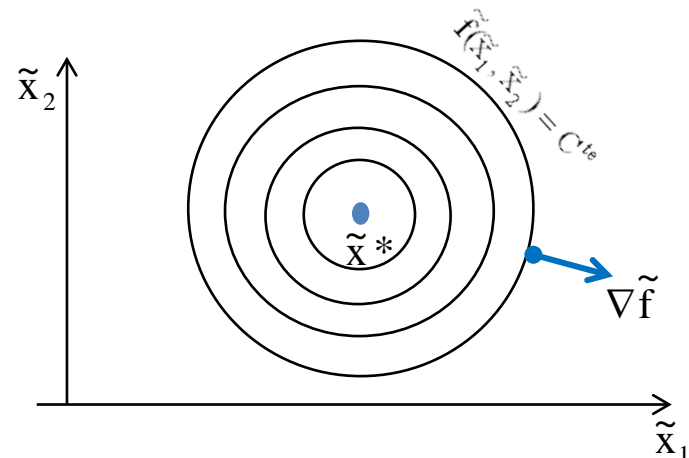
Préconditionnement de f

- Factorisation de Cholesky (si $H(x)$ définie positive) : $H(x) = LL^T$
- Conditionnement optimal (minimal) de f en x pour : $\tilde{x} = L^T x \Rightarrow \tilde{H}(\tilde{x}) = I \Rightarrow \kappa(\tilde{H}) = 1$

Sans conditionnement



Avec conditionnement



1.1.5 Système linéaire perturbé

Perturbation du second membre

- Système non perturbé : $Ax = b \rightarrow \text{solution } x^*$
- Système perturbé au 2nd membre : $A(x + \Delta x_b) = b + \Delta b \rightarrow \text{solution } x^* + \Delta x_b$
- Système perturbé au 1^{er} membre : $(A + \Delta A)(x + \Delta x_A) = b \rightarrow \text{solution } x^* + \Delta x_A$

$$\begin{cases} Ax^* = b \\ A(x^* + \Delta x_b) = b + \Delta b \\ (A + \Delta A)(x^* + \Delta x_A) = b \end{cases} \Rightarrow \begin{cases} Ax^* = b \\ A \cdot \Delta x_b = \Delta b \\ A \cdot \Delta x_A + \Delta A \cdot x^* = 0 \end{cases} \Rightarrow \begin{cases} b = Ax^* \\ \Delta x_b = A^{-1} \Delta b \\ \Delta x_A = -A^{-1} \cdot \Delta A \cdot x^* \end{cases}$$

- Majoration de la perturbation

$$b = Ax^* \Rightarrow \|b\| \leq \|A\| \cdot \|x^*\| \Rightarrow \frac{1}{\|x^*\|} \leq \frac{\|A\|}{\|b\|}$$

$$\begin{cases} \Delta x_b = A^{-1} \Delta b \\ \Delta x_A = -A^{-1} \cdot \Delta A \cdot x^* \end{cases} \Rightarrow \begin{cases} \|\Delta x_b\| \leq \|A^{-1}\| \cdot \|\Delta b\| \\ \|\Delta x_A\| \leq \|A^{-1}\| \cdot \|\Delta A\| \cdot \|x^*\| \end{cases} \Rightarrow \begin{cases} \frac{\|\Delta x_b\|}{\|x^*\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\Delta b\|}{\|b\|} \\ \frac{\|\Delta x_A\|}{\|x^*\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\Delta A\|}{\|A\|} \end{cases}$$

- Amplification maximale de la perturbation : $\kappa(A) = \|A\| \cdot \|A^{-1}\|$

1.1.5 Exemple

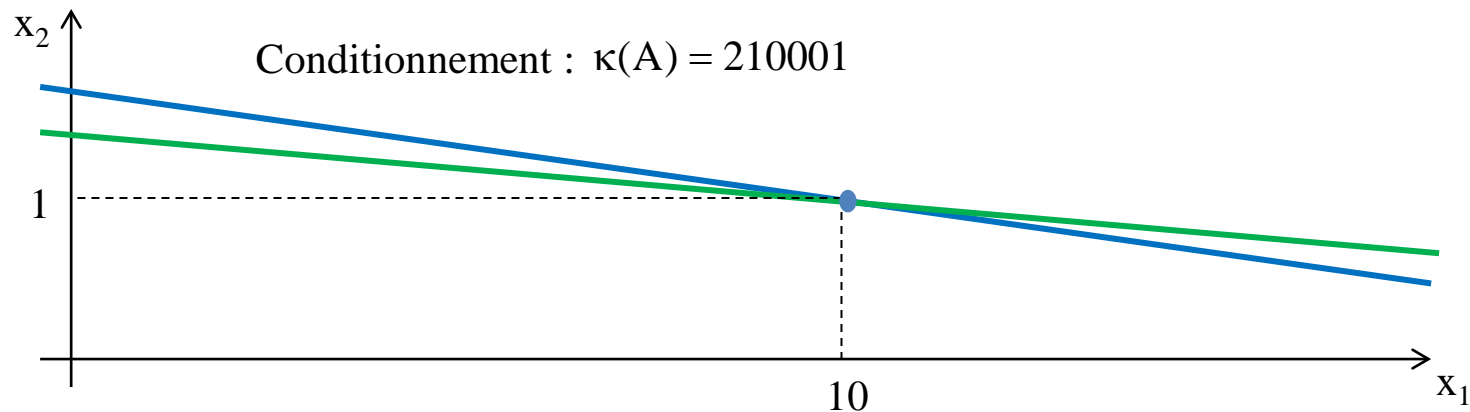
Système perturbé 2×2

- Système non perturbé

$$\begin{cases} 0.1x_1 + x_2 = 2 \\ 0.20002x_1 + 2x_2 = 4.0002 \end{cases} \Leftrightarrow \begin{pmatrix} 0.1 & 1 \\ 0.20002 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4.0002 \end{pmatrix} \Rightarrow \begin{cases} x_1 = 10 \\ x_2 = 1 \end{cases}$$

- Perturbation δA : $A = \begin{pmatrix} 0.101 & 1 \\ 0.20002 & 2 \end{pmatrix} \Rightarrow \begin{cases} x_1 = -0.101 \\ x_2 = 2.010 \end{cases}$

- Perturbation δb : $b = \begin{pmatrix} 2.01 \\ 4.0002 \end{pmatrix} \Rightarrow \begin{cases} x_1 = -990 \\ x_2 = 101.01 \end{cases}$



1.1.5 Mise à l'échelle

Principe

- Des valeurs numériques trop différentes sont sources de blocage des algorithmes.
Exemple : $(1 + 10^{-20}) - 1 = 1 - 1 = 0$ au lieu de 10^{-20} avec 16 chiffres significatifs
- Pour réduire les erreurs numériques, il faut que les différentes valeurs utilisées dans les calculs aient des ordres de grandeur comparables.
- Méthode de mise à l'échelle : transformation affine $X' = \alpha X + \beta$

Quantités à mettre à l'échelle

- Variables : $x \rightarrow x' \approx 1$ (\rightarrow déplacement sur toutes les composantes de x)
- Critère : $f \rightarrow f' \approx 1$ (\rightarrow tests d'arrêt sur variation de f)
- Contraintes : $c \rightarrow c' \approx 1$ (\rightarrow contraintes de poids équivalents)

- Jacobien : $\|\nabla c\| \rightarrow \|\nabla c'\| \approx 1$ (\rightarrow directions admissibles)
- Hessien : $\|\nabla^2 L\| \rightarrow \|\nabla^2 L'\| \approx 1$ (\rightarrow courbure, conditionnement)

Difficultés

- On ne peut pas simultanément mettre toutes les quantités à l'échelle \rightarrow choix expérimental
- Le facteur d'échelle dépend du point $x \rightarrow$ à adapter au cours des itérations
(mise à l'échelle dynamique)

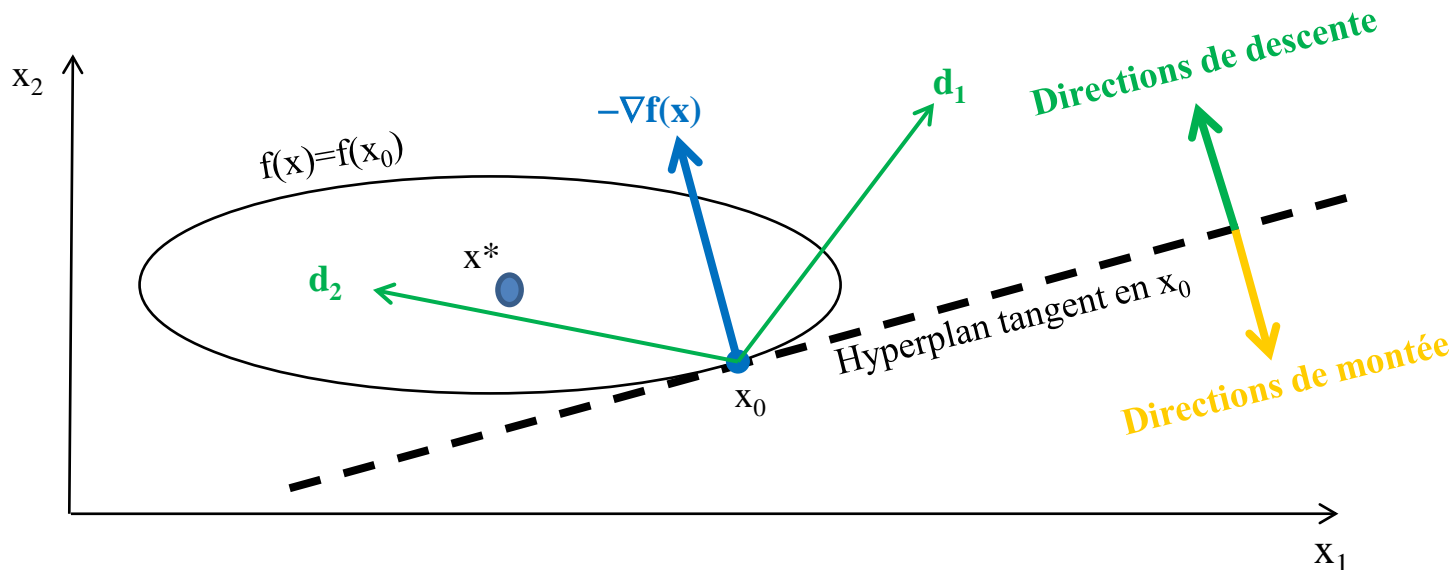
1.1.6 Direction de déplacement

- ☐ Direction de descente
 - Définition
 - Point de Newton
 - Point de Cauchy
- ☐ Direction admissible
- ☐ Contraintes linéaires

1.1.6 Direction de descente

Direction de descente

- Gradient de f en $x \in \mathbb{R}^n$: $g(x) = \nabla f(x)$
- Dérivée directionnelle de f en x suivant $d \in \mathbb{R}^n$: $f_d(x) = g(x)^T d$
 d est une **direction de descente** en x si : $f_d(x) = g(x)^T d < 0$
- La direction de **plus forte pente** d^+ est la direction du gradient : $d^+ = g(x)$
- La direction de **plus forte descente** d^- est opposée au gradient : $d^- = -g(x)$
 $\forall d \in \mathbb{R}^n / \|d\| = \|d^-\|, g(x)^T d \geq g(x)^T d^- = -\|g(x)\|^2$



1.1.6 Direction de descente

Variation suivant une direction

- Déplacement $d \in \mathbb{R}^n$ à partir de $x_0 \in \mathbb{R}^n$: $d = sd_0$ avec $d_0 \in \mathbb{R}^n$ = direction de déplacement
 $s \in \mathbb{R}$ = pas de déplacement suivant d_0

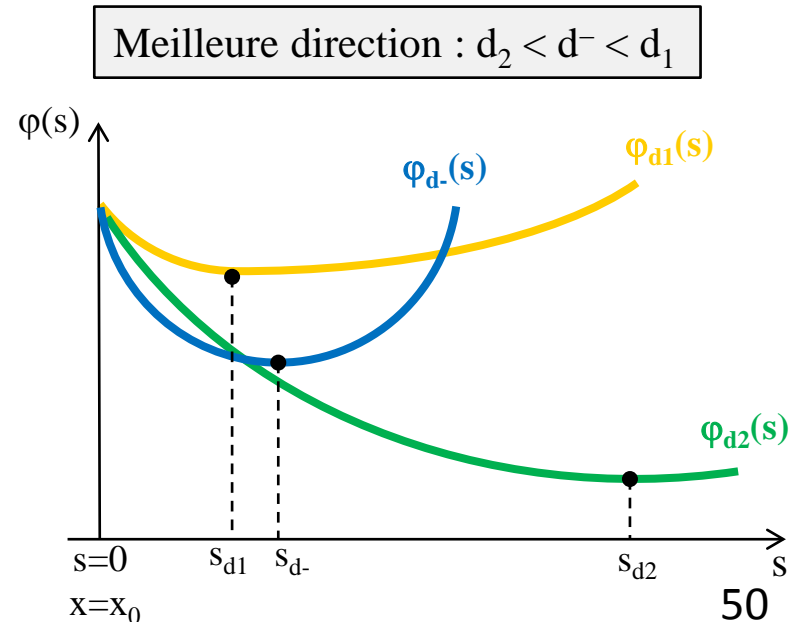
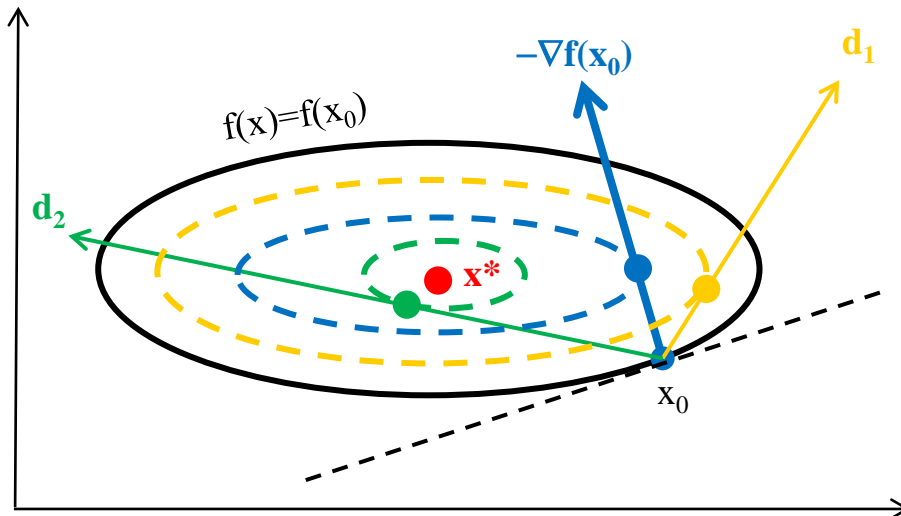
- Modèle quadratique au voisinage de x_0 :

$$\hat{f}(x_0 + sd_0) = f(x_0) + sg_0^T d_0 + \frac{1}{2}s^2 d_0^T H_0 d_0$$

$$\text{avec } \begin{cases} g_0 = \nabla f(x_0) \\ H_0 = \nabla^2 f(x_0) \end{cases}$$

$$\Rightarrow \hat{\varphi}(s) = \varphi(0) + s\varphi'(0) + \frac{1}{2}s^2\varphi''(0)$$

$$\text{avec } \begin{cases} \varphi'(0) = g_0^T d_0 \\ \varphi''(0) = d_0^T H_0 d_0 \end{cases}$$



1.1.6 Direction de descente

Minimisation locale

Deux points particuliers sont définis à partir du modèle quadratique de f en x_0 :

- Point de Newton : minimisation de f par rapport à $d \in \mathbb{R}^n \rightarrow x_n = x_0 + d_n$
- Point de Cauchy : minimisation de f suivant $d_0 = -g_0 \rightarrow x_c = x_0 - s_c g_0$

Point de Newton

$$\min_{d \in \mathbb{R}^n} \hat{f}(x_0 + d) = f(x_0) + g_0^T d + \frac{1}{2} d^T H_0 d$$

$$\Rightarrow d_n = -H_0^{-1} g_0 \text{ si } H_0 > 0$$

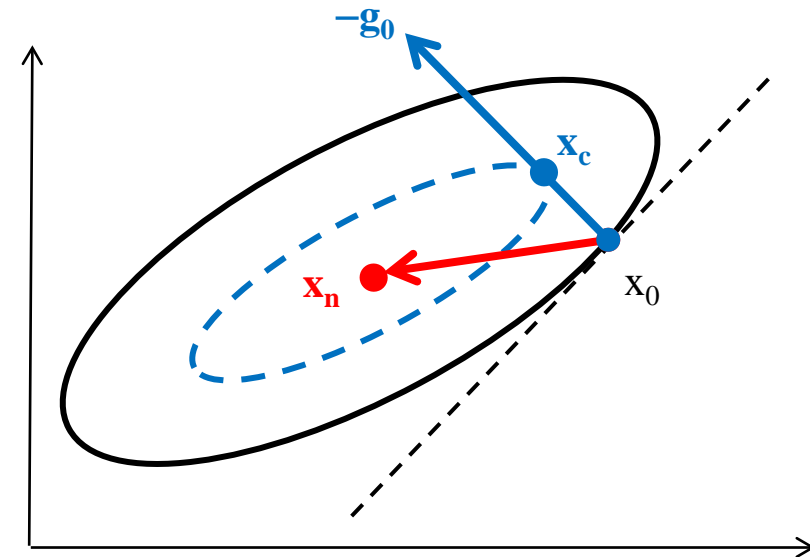
x_n existe si $H_0 = \nabla^2 f(x_0)$ est définie positive.

Point de Cauchy

$$\min_{s \in \mathbb{R}} \hat{\varphi}(x_0 - s g_0) = \varphi(0) - s \varphi'(0) + \frac{1}{2} s^2 \varphi''(0)$$

$$\Rightarrow s_c = \frac{\varphi'(0)^2}{\varphi''(0)} = \frac{(g_0^T g_0)^2}{g_0^T H_0 g_0} \text{ si } \varphi''(0) = g_0^T H_0 g_0 > 0$$

x_c existe si f est convexe suivant g_0 (condition moins forte que H_0 définie positive).



1.1.6 Direction admissible

Direction admissible

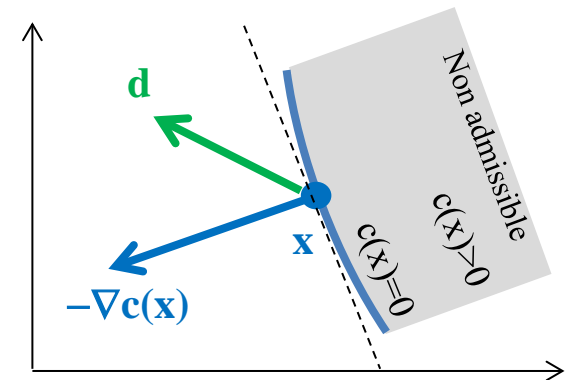
$d \in \mathbb{R}^n$ direction de déplacement à partir de $x \in X_{\text{adm}}$ point admissible

Définition : d direction admissible

$$\Leftrightarrow \exists \varepsilon > 0 / \forall s, 0 < s \leq \varepsilon \Rightarrow x + sd \in X_{\text{adm}}$$

On peut se déplacer d'au moins ε suivant d à partir de x en restant admissible

- Contrainte égalité : $\nabla c_E(x)^T d = 0 \rightarrow$ tangent
- Contrainte inégalité : $\nabla c_I(x)^T d \leq 0 \rightarrow$ intérieur

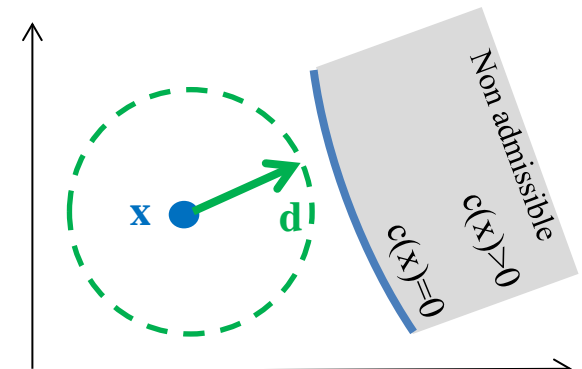


Ensemble convexe

X_{adm} convexe, $y \neq x, x, y \in X_{\text{adm}} \Rightarrow [x, y] \subset X_{\text{adm}}$
 $\Rightarrow d = y - x$ est une direction admissible à partir de x

Point intérieur

x point intérieur à X_{adm}
 \Rightarrow Toute direction $d \in \mathbb{R}^n$ est admissible à partir de x



1.1.6 Contraintes linéaires

Contraintes linéaires

Contraintes linéaires sous forme standard : $\begin{cases} Ax = b \\ x \geq 0 \end{cases}$
 $X_{\text{adm}} = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$

Direction admissible

$d \in \mathbb{R}^n$ direction admissible à partir de x point admissible $\Leftrightarrow \begin{cases} Ad = 0 \\ d_i \geq 0 \text{ si } x_i = 0 \end{cases}$

Preuve :

Pour $s > 0$ petit, on doit avoir $(x + sd) \in X_{\text{adm}} \Leftrightarrow \begin{cases} A(x + sd) = b \\ x + sd \geq 0 \end{cases} \Leftrightarrow \begin{cases} Ad = 0 \\ x + sd \geq 0 \end{cases}$ car $Ax = b$

Si $x_i > 0$, alors $x_i + sd_i > 0$ pour s assez petit

Si $x_i = 0$, alors $x_i + sd_i \geq 0$ si $d_i \geq 0$

Combinaison de directions admissibles

Toute combinaison linéaire à coefficients positifs de directions admissibles est une direction admissible.

Preuve : Une combinaison linéaire à coefficients positifs vérifie également $\begin{cases} Ad = 0 \\ d_i \geq 0 \text{ si } x_i = 0 \end{cases}$

Sommaire

- 1. Bases théoriques**
 - 1.1 Définitions
 - 1.2 Contraintes linéaires**
 - 1.2.1 Rappels d'algèbre linéaire
 - 1.2.2 Direction admissible
 - 1.2.3 Réduction
 - 1.2.4 Projection
 - 1.3 Contraintes non linéaires
 - 1.4 Conditions d'optimalité
- 2. Optimisation sans contraintes
- 3. Optimisation avec contraintes

1.2.1 Rappels d'algèbre linéaire

- ❑ Application linéaire
 - Matrice d'une application linéaire
 - Espace nul
 - Espace image
- ❑ Matrice
 - Valeurs et vecteurs propres
 - Matrices particulières
 - Factorisations
- ❑ Système linéaire
 - Solutions
 - Contraintes redondantes

1.2.1 Application linéaire

Application linéaire

Une matrice $A \in \mathbb{R}^{m \times n}$ définit une application linéaire de \mathbb{R}^n dans \mathbb{R}^m : $x \in \mathbb{R}^n \mapsto y = Ax \in \mathbb{R}^m$

Espace nul (ou noyau)

- L'espace nul de A est défini par :
- Si A est non singulière :

$$\text{Ker}(A) = \{x \in \mathbb{R}^n / Ax = 0\}$$

$$\text{Ker}(A) = \{0\}$$

Espace image

- L'espace image de A est défini par :
- Si A est non singulière :

$$\text{Im}(A) = \{y = Ax, x \in \mathbb{R}^n\}$$

$$\text{Im}(A) = \mathbb{R}^m$$

Le rang de A est la dimension de $\text{Im}(A)$: $\text{rang}(A) = \dim(\text{Im}(A))$

Théorème fondamental de l'algèbre

$\text{Ker}(A)$ et $\text{Im}(A^T)$ sont supplémentaires dans \mathbb{R}^n : $\text{Ker}(A) \oplus \text{Im}(A^T) = \mathbb{R}^n$

Tout $x \in \mathbb{R}^n$ s'écrit de façon unique comme somme d'un élément x_Z de $\text{Ker}(A)$
 et d'un élément x_Y de $\text{Im}(A^T)$

$$\forall x \in \mathbb{R}^n, x = x_Z + x_Y \quad \text{avec} \quad \begin{cases} x_Z \in \text{Ker}(A) \\ x_Y \in \text{Im}(A^T) \end{cases} \quad \text{de façon unique}$$

1.2.1 Matrice

Valeurs et vecteurs propres

Une matrice $A \in \mathbb{R}^{n \times n}$ admet la valeur propre $\sigma \in \mathbb{R}$
 s'il existe un vecteur non nul $v \in \mathbb{R}^n$ tel que : $Av = \sigma v$
 v est un vecteur propre associé à la valeur propre σ .

Matrices particulières

- A non singulière \Leftrightarrow Aucune valeur propre de A n'est nulle.
- A **symétrique** $\Leftrightarrow A^T = A$
 $\Rightarrow A$ admet n valeurs propres réelles (distinctes ou non)
 $\Rightarrow A$ admet une base orthonormée de vecteurs propres
- A orthogonale $\Leftrightarrow AA^T = A^T A = I$
- A **semi-définie positive** $\Leftrightarrow v^T A v > 0$ pour tout $v \in \mathbb{R}^n$
- A **définie positive** $\Leftrightarrow v^T A v \geq 0$ pour tout $v \in \mathbb{R}^n$
- A symétrique définie positive $\Rightarrow A$ admet n valeurs propres réelles positives (distinctes ou non)

1.2.1 Exemple

Valeurs propres d'une matrice 2×2

- Matrice A : $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$
- Valeurs propres : $\det(A - \sigma I) = 0 \Rightarrow \begin{vmatrix} a_{11} - \sigma & a_{12} \\ a_{21} & a_{22} - \sigma \end{vmatrix} = 0 \Rightarrow (a_{11} - \sigma)(a_{22} - \sigma) - a_{12}a_{21} = 0$

$$\Rightarrow \sigma^2 - (a_{11} + a_{22})\sigma + a_{11}a_{22} - a_{12}a_{21} = 0$$

$$\Rightarrow \sigma = \frac{1}{2} \left(a_{11} + a_{22} \pm \sqrt{(a_{11} + a_{22})^2 - 4(a_{11}a_{22} - a_{12}a_{21})} \right)$$

- Exemple : $A = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix} \Rightarrow \begin{cases} \sigma_1 = 1001.60 \\ \sigma_2 = 0.39936 \end{cases}$

Conditionnement : $\kappa(A) = 2508$

1.2.1 Exemple

Diagonalisation matrice symétrique 2×2

- Matrice Q : $Q = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ avec $b = c$
- Base orthonormée de vecteurs propres : $v_1 = \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix}$, $v_2 = \begin{pmatrix} -\sin \alpha \\ \cos \alpha \end{pmatrix}$
- Matrice de passage orthogonale : $P = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$

$$P^T Q P = \begin{pmatrix} a \cos^2 \alpha + 2b \cos \alpha \sin \alpha + d \sin^2 \alpha & (d - a) \cos \alpha \sin \alpha + b(\cos^2 \alpha - \sin^2 \alpha) \\ (d - a) \cos \alpha \sin \alpha + b(\cos^2 \alpha - \sin^2 \alpha) & a \sin^2 \alpha - 2b \cos \alpha \sin \alpha + d \cos^2 \alpha \end{pmatrix}$$

→ diagonale si $(d - a) \cos \alpha \sin \alpha + b(\cos^2 \alpha - \sin^2 \alpha) = 0$

avec $b \neq 0$ sinon Q directement diagonale $\Rightarrow \cos \alpha \neq 0$

- Direction des axes principaux (vecteurs propres) : $\tan^2 \alpha + \frac{a - d}{b} \tan \alpha - 1 = 0$
 → 2 solutions α_1 et $\alpha_2 = \alpha_1 + \pi/2$

1.2.1 Permutation

Permutation de colonnes

$A \in \mathbb{R}^{m \times n}$, matrice à m lignes, n colonnes

$$A = \begin{matrix} & \begin{matrix} 1 & \dots & j & \dots & k & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ m \end{matrix} & \begin{pmatrix} \times & \dots & \times & \dots & \times & \dots & \times \\ \vdots & & \vdots & & \vdots & & \vdots \\ \times & \dots & \times & \dots & \times & \dots & \times \end{pmatrix} \end{matrix}$$

$E \in \mathbb{R}^{n \times n}$, matrice de permutation des colonnes j et k
 $E^T = E$

$$E = \begin{matrix} & \begin{matrix} 1 & \dots & j & \dots & k & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ j \\ \vdots \\ k \\ \vdots \\ n \end{matrix} & \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 1 & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \end{matrix}$$



$$AE = \begin{matrix} & \begin{matrix} 1 & \dots & j & \dots & k & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ m \end{matrix} & \begin{pmatrix} \times & \dots & \times & \dots & \times & \dots & \times \\ \vdots & & \vdots & & \vdots & & \vdots \\ \times & \dots & \times & \dots & \times & \dots & \times \end{pmatrix} \end{matrix}$$

1.2.1 Factorisation

Factorisation de matrice

$A \in \mathbb{R}^{m \times n}$, matrice à m lignes, n colonnes avec $m < n$

A de rang plein : $\text{rang}(A) = m < n$

$$A = \begin{matrix} & \begin{matrix} 1 & \dots & j & \dots & k & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ m \end{matrix} & \begin{pmatrix} \times & \dots & \times & \dots & \times & \dots & \times \\ \vdots & & \vdots & & \vdots & & \vdots \\ \times & \dots & \times & \dots & \times & \dots & \times \end{pmatrix} \end{matrix}$$

3 types de factorisations sont utiles dans les algorithmes d'optimisation :

- Factorisation **LU** → Pour réduire le problème (variables dépendantes et indépendantes)
Pour construire une base de l'espace nul
- Factorisation **QR** → Pour réduire le problème (variables dépendantes et indépendantes)
Pour construire une base orthogonale de l'espace nul
- Factorisation **LL^T** → Pour une matrice définie positive
ou **LDL^T** Pour rendre le hessien défini positif

1.2.1 Factorisation LU

Factorisation LU

$A \in \mathbb{R}^{m \times n}$, matrice à m lignes, n colonnes

$$A = \begin{matrix} & \begin{matrix} 1 & \dots & j & \dots & k & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ m \end{matrix} & \begin{pmatrix} \times & \dots & \times & \dots & \times & \dots & \times \\ \vdots & & \vdots & & \vdots & & \vdots \\ \times & \dots & \times & \dots & \times & \dots & \times \end{pmatrix} \end{matrix}$$

- Matrice carrée $n \times n$

$$AE = LU$$

L $n \times n$ triangulaire inférieure

U $n \times n$ triangulaire supérieure

- Matrice rectangulaire $m \times n$, $m < n \rightarrow$ Factorisation de A^T

$$EA^T = LU = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} U$$

L_1 $m \times m$ triangulaire inférieure

L_2 $(n-m) \times m$ pleine

U $m \times m$ triangulaire supérieure

- Base de l'espace nul :

$$Z = E^T \begin{pmatrix} L_1^{-T} L_2^T \\ -I \end{pmatrix} U^{-T}$$

Méthode de factorisation LU

\rightarrow Méthode d'élimination de Gauss (ou **méthode du pivot de Gauss**)

1.2.1 Factorisation QR

Factorisation QR

$A \in \mathbb{R}^{m \times n}$, matrice à m lignes, n colonnes

$$A = \begin{matrix} & \begin{matrix} 1 & \dots & j & \dots & k & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ m \end{matrix} & \begin{pmatrix} \times & \dots & \times & \dots & \times & \dots & \times \\ \vdots & & \vdots & & \vdots & & \vdots \\ \times & \dots & \times & \dots & \times & \dots & \times \end{pmatrix} \end{matrix}$$

- Matrice rectangulaire $m \times n$

$$\boxed{AE = QR}$$

$$\begin{array}{ll} Q & m \times m \text{ orthogonale} \\ R & m \times n \text{ triangulaire supérieure} \end{array} \rightarrow QQ^T = I$$

- Base de l'espace nul \rightarrow Factorisation de A^T

$$\begin{array}{ll} Q_1 & n \times m \text{ orthogonale} \\ Q_2 & n \times (n-m) \text{ orthogonale} \\ R & n \times m \text{ triangulaire supérieure} \end{array} \rightarrow Z = E^T(Q_2)$$

$$EA^T = QR = (Q_1 \ Q_2)R$$

Méthode de factorisation QR

\rightarrow **Méthode de Householder** ou méthode de Givens

1.2.1 Factorisation LL^T

Factorisation LL^T

$A \in \mathbb{R}^{n \times n}$, matrice carrée $n \times n$ symétrique définie positive

$$\boxed{E^T A E = L L^T} \quad \text{avec } L \text{ matrice } n \times n \text{ triangulaire inférieure}$$

- Lien avec la factorisation QR

$$A E = Q R \Rightarrow E^T A^T A E = R^T Q^T Q R = R^T R \quad \text{car } Q \text{ orthogonale}$$

$$\rightarrow \begin{cases} R = L \\ Q = A E R^{-1} \end{cases}$$

Méthode de factorisation LL^T ou LDL^T

- **Méthode de Cholesky**
 - Permet de vérifier que la matrice A est bien définie positive
- Méthode de Cholesky modifiée
 - Permet de rendre la matrice A définie positive en la modifiant au cours de la factorisation

1.2.1 Exemple

Factorisation LDL^T d'une matrice 3×3

$$\begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & l_{21} & l_{31} \\ 0 & 1 & l_{32} \\ 0 & 0 & 1 \end{pmatrix}$$

- Résolution directe

$$\begin{cases} a_{11} = d_1 \\ a_{21} = d_1 l_{21} \\ a_{31} = d_1 l_{31} \\ a_{22} = d_1 l_{21}^2 + d_2 \\ a_{32} = d_1 l_{31} l_{21} + d_2 l_{32} \\ a_{33} = d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 \end{cases} \Rightarrow \begin{cases} d_1 = a_{11} \\ l_{21} = \frac{a_{21}}{d_1} \\ l_{31} = \frac{a_{31}}{d_1} \\ d_2 = a_{22} - d_1 l_{21}^2 \\ l_{32} = \frac{a_{32} - d_1 l_{31} l_{21}}{d_2} \\ d_3 = a_{33} - d_1 l_{31}^2 - d_2 l_{32}^2 \end{cases}$$

$\rightarrow d_1, d_2, d_3 > 0$
 si A est définie positive

 \rightarrow sinon on modifie d_j
 en cours de factorisation
 $d_j = \max(\delta, d_j), \delta > 0$

1.2.1 Exemple

Factorisation LDL^T d'une matrice 2×2

$$\begin{pmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} 1 & l_{21} \\ 0 & 1 \end{pmatrix} \Rightarrow \begin{cases} d_1 = a_{11} \\ l_{21} = \frac{a_{21}}{d_1} \\ d_2 = a_{22} - d_1 l_{21}^2 \end{cases}$$

- Exemple

$$A = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix} \Rightarrow L = \begin{pmatrix} 1 & 0 \\ -0.4988 & 1 \end{pmatrix}, D = \begin{pmatrix} 802 & 0 \\ 0 & 0.4988 \end{pmatrix}$$

Les éléments de la matrice diagonale D sont positifs $\rightarrow A$ est définie positive

Valeurs propres de A :

$$\begin{cases} \sigma_1 = 1001.60 \\ \sigma_2 = 0.39936 \end{cases}$$

1.2.1 Factorisation LDL^T

Méthode de Cholesky modifiée

A matrice $n \times n$ symétrique \rightarrow matrice A' définie positive « proche » de A
 ($A' = A$ si A est définie positive)

$A' = LDL^T$ avec L triangulaire inférieure, D diagonale positive

- Notations : $A=(a_{ij})$, $L=(l_{ij})$, $D=(d_{ij})$, $i,j=1,\dots,n$

Algorithme

Pour $j = 1, 2, \dots, n$

$$c_{jj} = a_{jj} - \sum_{s=1}^{j-1} d_s l_{js}^2$$

$$d_j = c_{jj}$$

Si $d_j \leq \delta$

Modification pour rendre $d_j > 0$
 (si A n'est pas définie positive)

$$d_j = \max \left(|c_{jj}|, \left(\frac{\theta_j}{\beta} \right)^2, \delta \right) \text{ avec } \theta_j = \max_{j < i \leq n} |c_{ij}|$$

Pour $i = j+1, \dots, n$

$$c_{ij} = a_{ij} - \sum_{s=1}^{j-1} d_s l_{is} l_{js}$$

$$l_{ij} = \frac{c_{ij}}{d_j}$$

Fin boucle i

Fin boucle j

$$\Rightarrow \begin{cases} d_j \geq \delta \\ m_{ij} = l_{ij} \sqrt{d_j} \leq \beta, \quad i = j+1, \dots, n \end{cases}$$

Paramètres : $\delta > 0$ petit ($\delta = 10^{-3}$)
 $\beta > 0$ ($\beta = 1$)

1.2.1 Système linéaire

Système linéaire

$$Ax = b \quad \text{avec} \quad \begin{cases} A \in \mathbb{R}^{m \times n} & \rightarrow \text{matrice de rang } r : \text{rang}(A) = r \\ b \in \mathbb{R}^m & \rightarrow m \text{ équations} \\ x \in \mathbb{R}^n & \rightarrow n \text{ inconnues} \end{cases}$$

Le rang de A est la dimension du sous-espace engendré par A (image de A)

$$\text{Im}(A) = \{y = Ax, x \in \mathbb{R}^n\} \quad \rightarrow r = \dim(\text{Im}(A)) \leq m, n$$

Solutions possibles

- Pas de solution : système **incompatible** ($m > n$: plus d'équations que d'inconnues)
- Solution unique : système **non singulier** ($m = n$: autant d'équations que d'inconnues)
- Infinité de solutions : système **sous-déterminé** ($m < n$: moins d'équations que d'inconnues)

Problème d'optimisation

Contraintes linéaires $Ax = b$ \rightarrow système sous-déterminé (**$m < n$**)
 \rightarrow $n - m$ inconnues «libres» permettant de minimiser le critère

1.2.1 Système linéaire

Contraintes redondantes

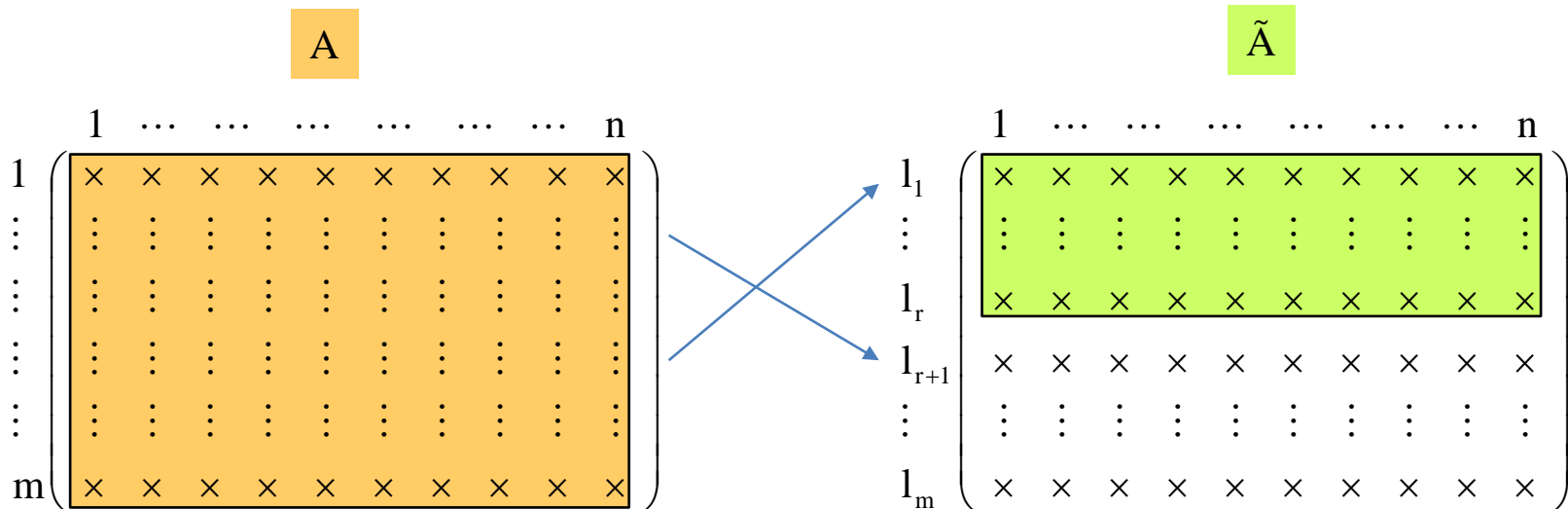
Pour un système sous-déterminé ($m < n$), si A est de rang déficient : $\text{rang}(A) = r < m$, on peut extraire de A une sous-matrice $\tilde{A} \in \mathbb{R}^{r \times n}$ de rang plein, telle que :

$$\tilde{A}x = \tilde{b} \Leftrightarrow Ax = b$$

\tilde{A} est composée des lignes l_1, \dots, l_r de A

Les lignes l_{r+1}, \dots, l_m sont combinaisons linéaires des lignes l_1, \dots, l_r .

Elles correspondent à des contraintes redondantes et peuvent être éliminées de la résolution.



1.2.2 Direction admissible

- ☐ Polytope
- ☐ Forme standard
- ☐ Sommet
- ☐ Base
- ☐ Solution de base
- ☐ Direction de base

1.2.2 Polytope

Définition

Polytope P dans \mathbb{R}^n

$$\mathbf{P} = \left\{ \mathbf{x} \in \mathbf{R}^n \mid \mathbf{Ax} \leq \mathbf{b} \right\}$$

$$\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$$

Interprétation géométrique

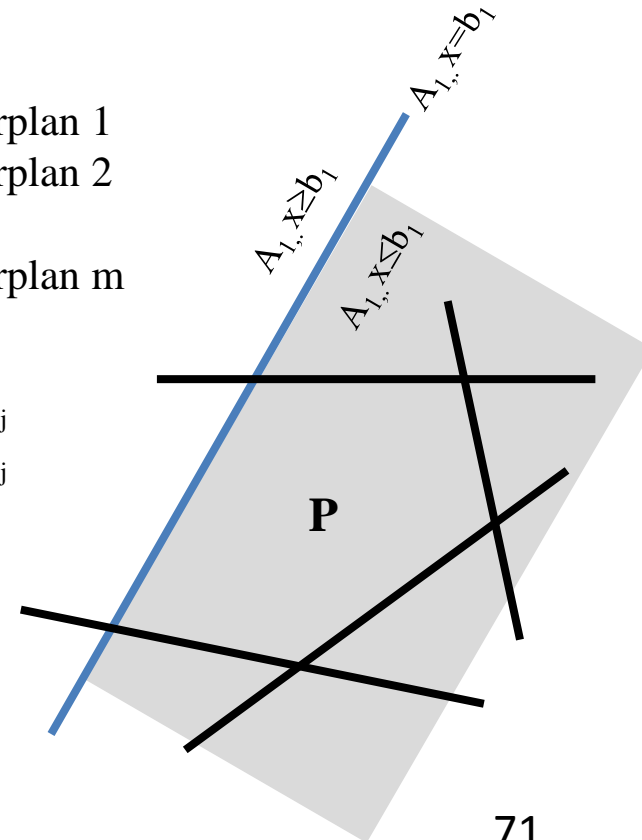
[illegible]

Chaque hyperplan j sépare \mathbb{R}^n en 2 sous-espaces : $\begin{cases} \mathbf{A}_{j, \cdot} \mathbf{x} \leq \mathbf{b}_j \\ \mathbf{A}_{j, \cdot} \mathbf{x} \geq \mathbf{b}_j \end{cases}$

P = ensemble de points de \mathbb{R}^n délimité par m hyperplans

→ Polytope dans \mathbb{R}^2 = **polygone**

→ Polytope dans \mathbb{R}^3 = **polyèdre**



1.2.2 Forme standard

Forme standard

Polytope P dans \mathbb{R}^n sous forme standard

$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$$

$$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

A de rang plein (élimination contraintes redondantes $\rightarrow \tilde{A}$)

Passage sous forme standard

- Contrainte inégalité : Transformation en contrainte égalité
 Ajout d'une **variable d'écart** positive

$$c(x) \leq b \Leftrightarrow \begin{cases} c(x) + z = b \\ z \geq 0 \end{cases} \Leftrightarrow \begin{cases} c'(x) = b \\ z \geq 0 \end{cases} \text{ avec } c'(x) = c(x) + z$$

$$c(x) \geq b \Leftrightarrow \begin{cases} c(x) - z = b \\ z \geq 0 \end{cases} \Leftrightarrow \begin{cases} c'(x) = b \\ z \geq 0 \end{cases} \text{ avec } c'(x) = c(x) - z$$

- Contraintes de bornes : Changement de variable \rightarrow borne inférieure
 Ajout d'une **variable d'écart** positive \rightarrow borne supérieure

$$x_l \leq x \leq x_u \Leftrightarrow 0 \leq x - x_l \leq x_u - x_l \Leftrightarrow \begin{cases} x' = x - x_l, x' \geq 0 \\ x' \leq x_u - x_l \end{cases} \Leftrightarrow \begin{cases} x' = x - x_l, x' \geq 0 \\ x' + z = x_u - x_l, z \geq 0 \end{cases}$$

- Variable libre : Différence de 2 variables positives

$$x \in \mathbb{R} \Leftrightarrow x = z - y, y, z \geq 0$$

1.2.2 Exemple

Mise sous forme standard

- Problème linéaire (P)

$$\min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3 \quad \text{sous} \quad \begin{cases} -x_1 + 3x_2 = 5 \\ 2x_1 - x_2 + 3x_3 \geq 6 \\ x_1 \in \mathbf{R}, x_2 \geq 1, x_3 \leq 4 \end{cases}$$

- Changement de variables pour les bornes

$$\begin{cases} x_1 \in \mathbf{R} \\ x_2 \geq 1 \\ x_3 \leq 4 \end{cases} \Leftrightarrow \begin{cases} x_1 = z_1 - y_1 \rightarrow y_1, z_1 \geq 0 \\ x_2' = x_2 - 1 \rightarrow x_2' \geq 0 \\ x_3' = 4 - x_3 \rightarrow x_3' \geq 0 \end{cases}$$

- Variables d'écart pour les contraintes inégalité

$$2x_1 - x_2 + 3x_3 \geq 6 \Leftrightarrow 2x_1 - x_2 + 3x_3 - z_2 = 6 \rightarrow z_2 \geq 0$$

- Problème équivalent à (P) sous forme standard

$$\min_{y_1, z_1, z_2, x_2', x_3'} z_1 - y_1 + 2x_2' - 3x_3' + 14 \quad \text{sous} \quad \begin{cases} y_1 - z_1 + 3x_2' = 2 \\ 2z_1 - 2y_1 - x_2' - 3x_3' - z_2 = -5 \end{cases}$$

1.2.2 Sommet

Sommet

Polytope P dans \mathbb{R}^n sous forme standard

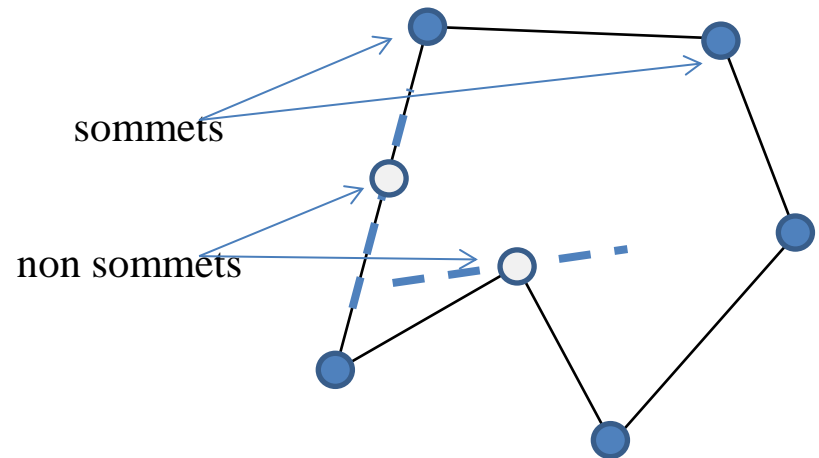
$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \quad \begin{array}{l} A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \\ A \text{ de rang plein : } \text{rang}(A)=r=m \leq n \end{array}$$

Définition

$x \in P$ est un sommet de P

\Leftrightarrow

On ne peut pas trouver $y, z \in P$, différents de x
tels que x soit combinaison convexe de y et z
i.e. $x = \lambda y + (1 - \lambda)z$ avec $0 < \lambda < 1$



Existence

Tout polytope non vide possède au moins un sommet.

1.2.2 Base

Base

Polytope P dans \mathbb{R}^n sous forme standard

$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

- A est de rang plein $r=m \leq n \Rightarrow$ Il existe m colonnes indépendantes
- On choisit une sous-matrice $B \in \mathbb{R}^{m \times m}$ de rang plein (parmi C_n^m combinaisons possibles)

$$AE = \begin{pmatrix} m & n-m \\ B & N \end{pmatrix} = \begin{matrix} & \begin{matrix} \text{B} & & \text{N} \end{matrix} \\ & \begin{matrix} 1 & \dots & m & m+1 & \dots & k-1 & k & k+1 & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ m \end{matrix} & \begin{pmatrix} \times & \dots & \times & \times & \dots & \times & \times & \times & \dots & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & \dots & \times & \times & \dots & \times & \times & \times & \dots & \times \end{pmatrix} \end{matrix}$$

E matrice de permutation des colonnes de A : $EE^T = I$

$A_{.,k} = k^{\text{ème}}$ colonne de AE

- $B = \text{matrice de base} \rightarrow B \in \mathbb{R}^{m \times m}$ inversible
- $N = \text{matrice hors base} \rightarrow N \in \mathbb{R}^{m \times (n-m)}$

1.2.2 Solution de base

Identification des sommets

Polytope P dans \mathbb{R}^n sous forme standard

$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

- Choix d'une base $B \in \mathbb{R}^{m \times m}$

$$AE = \begin{pmatrix} m & n-m \\ B & N \end{pmatrix} \quad E^T x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$$

- $x_B \in \mathbb{R}^m$ = variables en base (ou liées ou dépendantes)
 $x_N \in \mathbb{R}^{n-m}$ = variables hors base (ou libres ou indépendantes)

- **Point admissible** : $x \in P \Leftrightarrow \begin{cases} Ax = b \\ x \geq 0 \end{cases} \Leftrightarrow \begin{cases} Bx_B + Nx_N = b \\ x \geq 0 \end{cases} \Leftrightarrow \boxed{\begin{cases} x_B = B^{-1}(b - Nx_N) \\ x \geq 0 \end{cases}}$

- Identification des sommets

Tout point x tel que : $\begin{cases} x_B = B^{-1}b \geq 0 \\ x_N = 0 \end{cases} \Rightarrow E^T x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ est un sommet du polytope.

→ « **solutions de base** »

1.2.2 Solution de base

Identification des sommets

Preuve : par l'absurde

- On suppose le point $x : E^T x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ avec $\begin{cases} x_B = B^{-1}b \geq 0 \\ x_N = 0 \end{cases}$ n'est pas un sommet de P .
 x peut alors s'écrire : $x = \lambda y + (1 - \lambda)z$ avec $0 < \lambda < 1$
 y et $z \in P$, $y \neq x$, $z \neq x$
- En décomposant suivant les composantes B et N : $\begin{cases} x_B = \lambda y_B + (1 - \lambda)z_B \\ x_N = \lambda y_N + (1 - \lambda)z_N \end{cases}$ avec $\begin{cases} y \in P \Rightarrow y_N \geq 0 \\ z \in P \Rightarrow z_N \geq 0 \end{cases}$
- A partir de $x_N = 0$
 $x_N = \lambda y_N + (1 - \lambda)z_N = 0$ avec $\begin{cases} y_N, z_N \geq 0 \\ 0 < \lambda < 1 \end{cases} \Rightarrow \begin{cases} y_N = 0 \\ z_N = 0 \end{cases}$
- A partir de $y \in P$: $Ay = b \Leftrightarrow By_B + Ny_N = b \Rightarrow y_B = B^{-1}b$
 A partir de $z \in P$: $Az = b \Leftrightarrow Bz_B + Nz_N = b \Rightarrow z_B = B^{-1}b$
 $\Rightarrow \begin{cases} y_B = x_B \\ z_B = x_B \end{cases}$
- On obtient $y = z = x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ en contradiction avec l'hypothèse que x n'est pas un sommet de P

1.2.2 Solution de base

Solution de base

Polytope P dans \mathbb{R}^n sous forme standard

$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \quad \begin{array}{l} A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \\ A \text{ de rang plein : } \text{rang}(A) = r = m \leq n \end{array}$$

Définition

$x \in \mathbb{R}^n$ est une **solution de base** de P

\Leftrightarrow

Il existe m indices i_1, \dots, i_m tels que

- La matrice $B \in \mathbb{R}^{m \times m}$ composée des colonnes i_1, \dots, i_m de A est de rang plein
- Les $n-m$ composantes $x_i, i \neq i_1, \dots, i_m$ sont nulles $\rightarrow x_N = 0$
- x vérifie $Ax = b$ $\rightarrow x_B = B^{-1}b$ $\Rightarrow E^T x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$

Solution de base admissible

Une solution de base x est **admissible** si toutes ses composantes sont positives ($x \in P$).

x vérifie également $x \geq 0 \Rightarrow x_B = B^{-1}b \geq 0$

\rightarrow **Base admissible ou réalisable**, solution de base admissible ou réalisable

Solution de base dégénérée

Une solution de base x est **dégénérée** si plus de $n-m$ composantes de x sont nulles.

$x_N = 0$ par définition ($n-m$ composantes) $\Rightarrow x_B$ comporte des composantes nulles

1.2.2 Solution de base

Lien sommet – solution de base

Polytope P dans \mathbb{R}^n sous forme standard

$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

A de rang plein : $\text{rang}(A) = r = m \leq n$

- I = indices des composantes nulles en $x^* \in P$: $I^* = \{i / x_i^* = 0\}$
 (= contraintes inégalités actives)
- S = variété linéaire définie par : $S^* = \{x \in \mathbb{R}^n / Ax = b, x_i = 0, \forall i \in I^*\}$

$$x^* \in \mathbb{R}^n \text{ est un } \textbf{sommet} \text{ de } P \Leftrightarrow S^* = \{x^*\} \Leftrightarrow x^* \text{ est une } \textbf{solution de base admissible} \text{ de } P$$

Lien sommet – contraintes actives

$$x^* \in P \text{ est un } \textbf{sommet} \text{ de } P \Leftrightarrow \text{Au moins } \textbf{n} \text{ contraintes sont actives en } x^*$$

- m contraintes égalité : $Ax^* = b$
- $n-m$ contraintes inégalité : $x_N^* = 0$

Les m contraintes inégalité sur x_B peuvent être actives ou non :

$$x_B^* = B^{-1}b \geq 0 \rightarrow \text{dégénérescence}$$

1.2.2 Solution de base

Lien sommet – solution de base

Eléments de la démonstration : Le sens inverse est déjà démontré (identification des sommets)

Sens direct : $x \in \mathbb{R}^n$ est un sommet de $P \Rightarrow x$ est une solution de base admissible de P

- *On suppose par contraposée que x n'est pas une solution de base admissible.*
- *En décomposant suivant les composantes B et N : $AE = \begin{pmatrix} m & n-m \\ B & N \end{pmatrix} E^T x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$
avec B, N choisies telles que $x_B > 0$*
- *x n'est pas une solution de base admissible \Rightarrow Il existe au moins une composante $x_{Nk} \neq 0$*

*On construit la direction d^k $E^T d^k = \begin{pmatrix} d_B^k \\ d_N^k \end{pmatrix} = \begin{pmatrix} -B^{-1} A_{.,k} \\ 0 \end{pmatrix} + e_k = \begin{pmatrix} 1 & \dots & m & m+1 & \dots & k-1 & k & k+1 & \dots & n \\ d_1 & \dots & d_m & 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix}^T$
(dite $k^{\text{ème}}$ direction de base)*

$$Ad^k = Bd_B^k + Nd_N^k = -BB^{-1}A_{.,k} + Nd_{Nk} = -A_{.,k} + A_{.,k} = 0 \quad \text{car} \quad N(d_N^k)_k = A_{.,k}$$

$$\Rightarrow A(x + \alpha d^k) = Ax + \alpha Ad^k = b$$

avec $A_{.,k} = k^{\text{ème}}$ colonne de AE , d^k a toutes ses composantes hors base nulles sauf la $k^{\text{ème}} = 1$

- *Comme $\begin{cases} x_B > 0 \\ x_{Nk} > 0 \end{cases}$, on peut se déplacer suivant $-d^k$ et $+d^k$ d'un pas petit en conservant $\begin{cases} Ax = b \\ x \geq 0 \end{cases}$*

\rightarrow On obtient 2 points y et z de P tels que $x = \lambda y + (1 - \lambda)z$ avec $0 < \lambda < 1$

$\rightarrow x$ n'est pas un sommet de P .

1.2.2 Exemple

Recherche des solutions de base

Polytope P dans \mathbb{R}^4 sous forme standard

$$P = \{(x_1, x_2, x_3, x_4) / Ax = b, x \geq 0\} \quad \text{avec} \quad A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

- On utilise les contraintes pour réduire le problème à (x_1, x_2)

$$Ax = b \Leftrightarrow \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 - x_2 + x_4 = 1 \end{cases} \Leftrightarrow \begin{cases} x_3 = 1 - x_1 - x_2 \\ x_4 = 1 - x_1 + x_2 \end{cases}$$

$$x \geq 0 \Rightarrow \begin{cases} x_3 \geq 0 \\ x_4 \geq 0 \end{cases} \Rightarrow \begin{cases} x_1 + x_2 \leq 1 \\ x_1 - x_2 \leq 1 \end{cases}$$

- Polytope P' réduit dans \mathbb{R}^2

$$P' = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} / \begin{cases} x_1 + x_2 \leq 1 \\ x_1 - x_2 \leq 1 \end{cases}, \begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \end{cases} \right\}$$

- Les polytopes P dans \mathbb{R}^4 et P' dans \mathbb{R}^2 sont équivalents : $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \in P \Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in P'$

Remarque : P = forme standard de P'

1.2.2 Exemple

Recherche des solutions de base

- Représentation de P' dans \mathbb{R}^2

$$P' = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} / \begin{cases} x_1 + x_2 \leq 1 \\ x_1 - x_2 \leq 1 \end{cases}, \begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \end{cases} \right\}$$

→ représentation des valeurs possibles de (x_1, x_2)
 pour $(x_1, x_2, x_3, x_4) \in P$

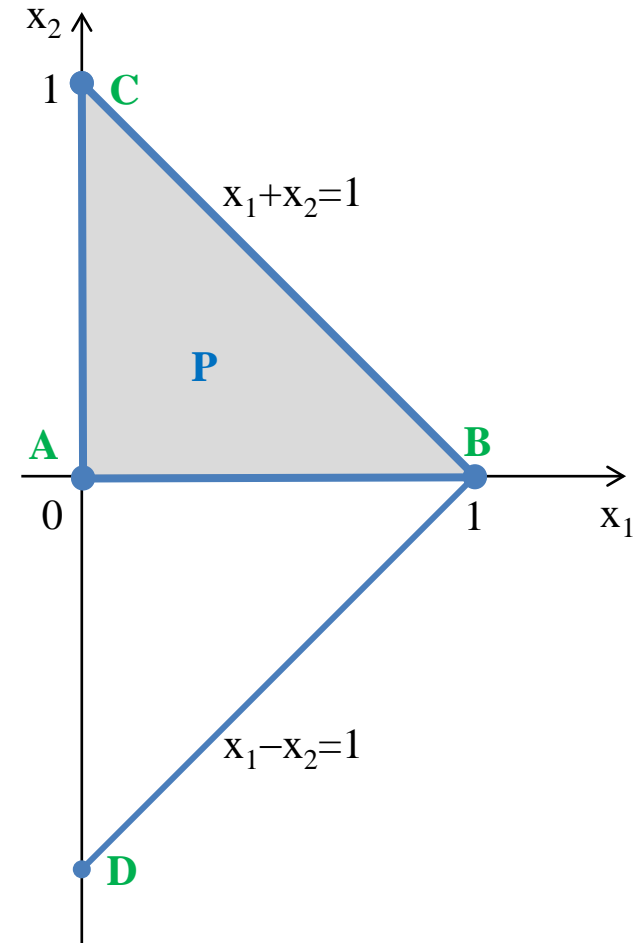
- Contraintes de P : $Ax = b \Leftrightarrow \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
 4 variables – 2 contraintes

- **Base de P**

→ choisir 2 colonnes indépendantes de A
 → 6 combinaisons possibles

- **Solution de base**

→ fixer les 2 variables hors base x_N à 0
 → calculer les 2 variables de base x_B pour vérifier $Ax=b$
 → base admissible si $x_B \geq 0$



1.2.2 Exemple

Recherche des solutions de base

Examen des 6 bases possibles de P

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad x_B = B^{-1}b$$

- **Base (x_1, x_2)** : $B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $B^{-1} = \begin{pmatrix} 0,5 & 0,5 \\ 0,5 & -0,5 \end{pmatrix}$, $x_B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

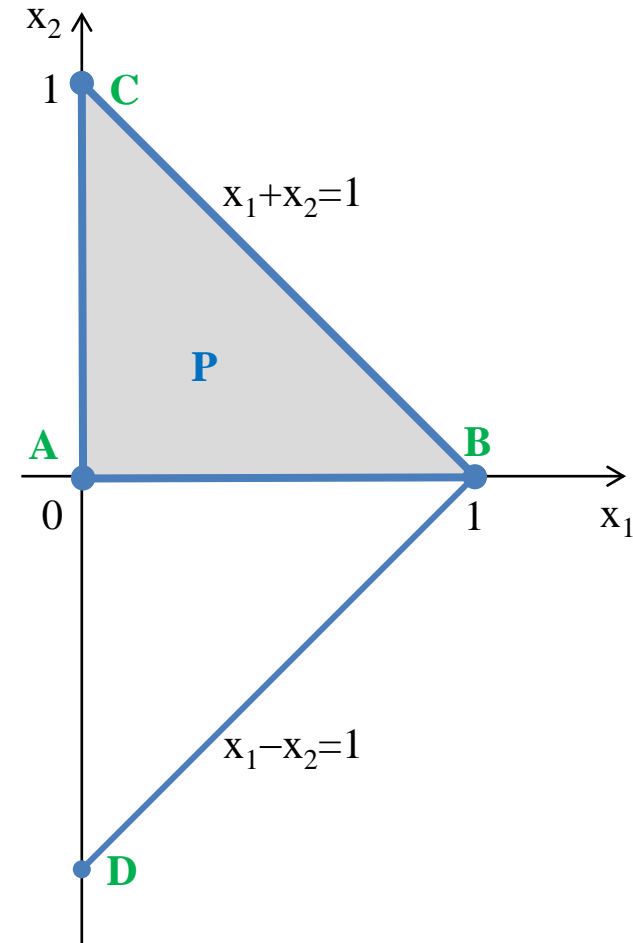
$x = (1 \ 0 \ 0 \ 0)$ admissible → **point B**

- **Base (x_1, x_3)** : $B = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, $B^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}$, $x_B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$x = (1 \ 0 \ 0 \ 0)$ admissible → **point B**

- **Base (x_1, x_4)** : $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $B^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$, $x_B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$x = (1 \ 0 \ 0 \ 0)$ admissible → **point B**



1.2.2 Exemple

Recherche des solutions de base

Examen des 6 bases possibles de P

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad x_B = B^{-1}b$$

- **Base (x_2, x_3)** : $B = \begin{pmatrix} 1 & 1 \\ -1 & 0 \end{pmatrix}$, $B^{-1} = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}$, $x_B = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$

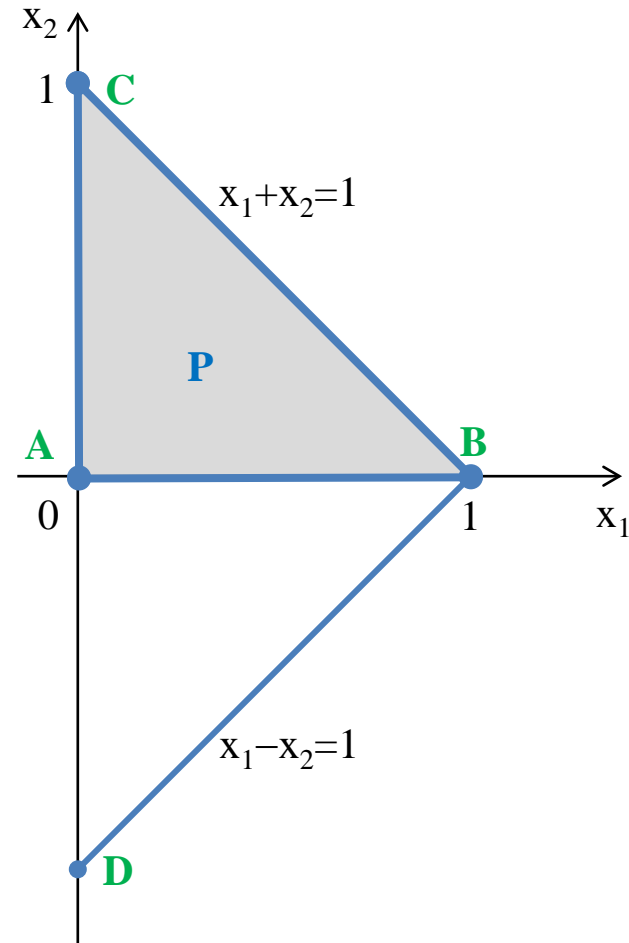
$x = (0 \ -1 \ 2 \ 0)$ non admissible → **point D**

- **Base (x_2, x_4)** : $B = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$, $B^{-1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $x_B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$x = (0 \ 1 \ 0 \ 2)$ admissible → **point C**

- **Base (x_3, x_4)** : $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $B^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $x_B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$x = (0 \ 0 \ 1 \ 1)$ admissible → **point A**



1.2.2 Direction de base

Direction de déplacement à partir d'un sommet

Polytope P dans \mathbb{R}^n sous forme standard

$$P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} \quad A \in \mathbb{R}^{m \times n} \text{ de rang plein, } b \in \mathbb{R}^m$$

$x \in \mathbb{R}^n$ solution de base admissible de P :

$$E^T x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} \geq 0$$

$d \in \mathbb{R}^n$ direction de déplacement :

$$E^T d = \begin{pmatrix} d_B \\ d_N \end{pmatrix}$$

Direction admissible

$$d \text{ direction admissible en } x \Leftrightarrow \begin{cases} Ad = 0 \\ d_i \geq 0 \text{ si } x_i = 0 \end{cases} \quad (\text{contraintes linéaires})$$

$$Ad = 0 \Leftrightarrow Bd_B + Nd_N = 0 \Leftrightarrow d_B = -B^{-1}Nd_N$$

$$d \text{ direction admissible en } x \Leftrightarrow \begin{cases} d_B = -B^{-1}Nd_N \\ d_N \geq 0 \text{ car } x_N = 0 \\ d_{Bi} \geq 0 \text{ si } x_{Bi} = 0 \end{cases} \quad (\text{solution de base dégénérée})$$

→ « **directions de base** »

1.2.2 Direction de base

Direction de base

$x \in \mathbb{R}^n$ solution de base admissible de P : $E^T x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} \geq 0$

k = indice d'une variable hors-base

$d^k = k^{\text{ème}}$ direction de base en x :

$$E^T d^k = \begin{pmatrix} d_B^k \\ d_N^k \end{pmatrix} \text{ noté } \begin{pmatrix} d_B \\ d_N \end{pmatrix}$$

- Les **composantes d_N** sur les variables hors base sont toutes nulles, sauf sur la variable x_k

$$E^T \begin{pmatrix} 0 \\ d_N \end{pmatrix} = e_k = \begin{pmatrix} 1 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & \dots & 0 \end{pmatrix}^T$$

- Les **composantes d_B** sur les variables en base vérifient la 1^{ère} condition de direction admissible

$$\begin{aligned} Ad = 0 &\Rightarrow Bd_B + Nd_N = 0 \Rightarrow d_B = -B^{-1}Nd_N = -B^{-1} \sum_{j \text{ hors base}} A_{.,j} d_j \\ &\Rightarrow d_B = -B^{-1}A_{.,k} \quad (A_{.,k} = k^{\text{ème}} \text{ colonne de AE}) \end{aligned}$$

- Définition**

La $k^{\text{ème}}$ direction de base en x est : $E^T d^k = \begin{pmatrix} d_B^k \\ d_N^k \end{pmatrix} = \begin{pmatrix} -B^{-1}A_{.,k} \\ 0 \end{pmatrix} + e_k = \begin{pmatrix} 1 & \dots & 0 & 1 & 0 & \dots & 0 \\ d_1 & \dots & d_m & 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix}^T$

- Interprétation géométrique : directions de base = arêtes du polytope en x

1.2.2 Direction de base

Direction de base admissible

$x \in \mathbb{R}^n$ solution de base admissible de P, k = indice d'une variable hors-base

La $k^{\text{ème}}$ direction de base d^k en x vérifie par définition :
$$\begin{cases} Ad = 0 \\ d_N \geq 0 \end{cases}$$

Pour que d^k soit une direction admissible, il faut également vérifier : $d_{Bi} \geq 0$ si $x_{Bi} = 0$

Cas d'une base non dégénérée

x solution de base admissible non dégénérée ($x_B > 0$)



Toutes les directions de base en x sont admissibles

Combinaison de directions de base

$x \in \mathbb{R}^n$ solution de base admissible de P

Toute direction admissible d en x est combinaison linéaire des directions de base d^k en x

$$d = \sum_{k \text{ hors base}} \alpha_k d^k \quad \text{avec } d^k = k^{\text{ème}} \text{ direction de base en } x$$

1.2.2 Direction de base

Cas d'une base non dégénérée

Preuve : on suppose que x est une solution de base non dégénérée.

- La $k^{\text{ème}}$ direction de base d^k vérifie

$$Ad^k = Bd_B^k + Nd_N^k = -BB^{-1}A_{.,k} + Nd_{Nk} = -A_{.,k} + A_{.,k} = 0 \quad \text{car} \quad N(d_N^k)_k = A_{.,k}$$

- Comme $\begin{cases} x_B > 0 \\ d_N > 0 \end{cases}$, on peut se déplacer suivant d^k à partir de x en restant admissible $\rightarrow d^k$ est une direction admissible.

Combinaison de directions de base

Preuve : on suppose que d est une direction admissible.

- En décomposant suivant les composantes B et N : $Ad = Bd_B + Nd_N = 0 \Rightarrow d_B = -B^{-1}Nd_N$
- En notant d_k les composantes de d_N dans la base canonique de R^n : $d_N = \sum_{k \in N} d_k e_k$
- On obtient pour d : $d = \begin{pmatrix} d_B \\ d_N \end{pmatrix} = \sum_{k \in N} d_k \begin{pmatrix} B^{-1}A_{.,k} \\ e_k \end{pmatrix}$

$$\text{avec } d^k = \begin{pmatrix} B^{-1}A_{.,k} \\ e_k \end{pmatrix} = k^{\text{ème}} \text{ direction de base} \rightarrow d = \text{combinaison linéaire des } d^k$$

1.2.2 Exemple

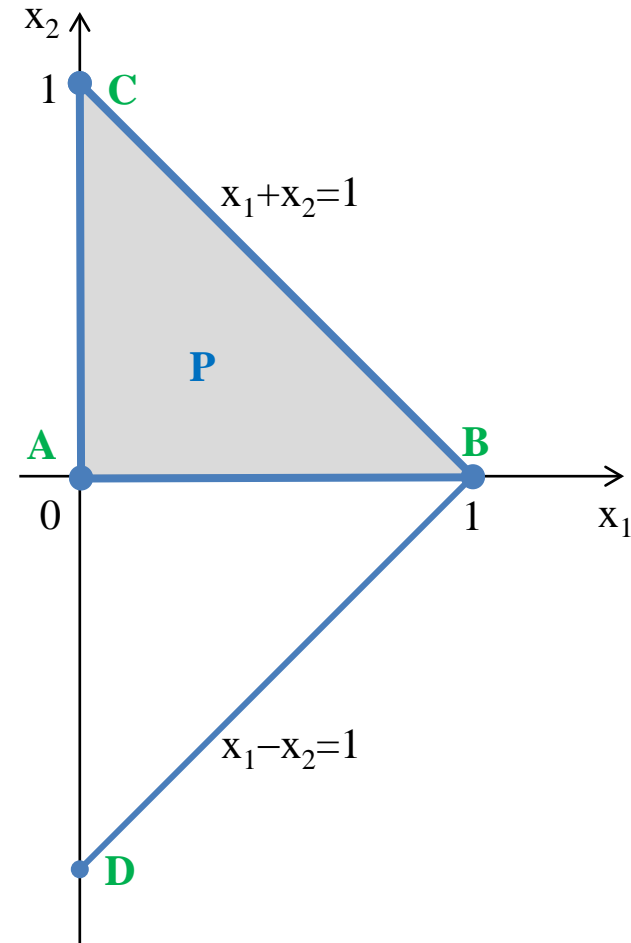
Recherche des directions de base

Polytope P dans \mathbb{R}^4 sous forme standard

$$P = \{(x_1, x_2, x_3, x_4) / Ax = b, x \geq 0\}$$

$$\text{avec } A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

- **Direction de base en une solution de base admissible**
 - choisir une variable hors base (k)
 - fixer la composante hors base correspondante d_{Nk} à 1
 - fixer les autres composantes hors base d_N à 0
 - calculer les composantes en base d_B par $-B^{-1}A_{.,k}$
- Si la base est non dégénérée, la direction est admissible. Sinon, il faut vérifier $d_B \geq 0$ sur les composantes $x_B = 0$
- Sommets de P
 - 2 variables hors base à chaque sommet
 - 2 directions de base (= arêtes du polytope)



1.2.2 Exemple

Recherche des directions de base

Examen de directions de base de P

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}$$

- **Base admissible (x_2, x_4)** : $x = (0 \ 1 \ 0 \ 2) \rightarrow$ **point C**

$$B = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}, B^{-1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- **Direction de base d^1** correspondant à la variable hors base x_1

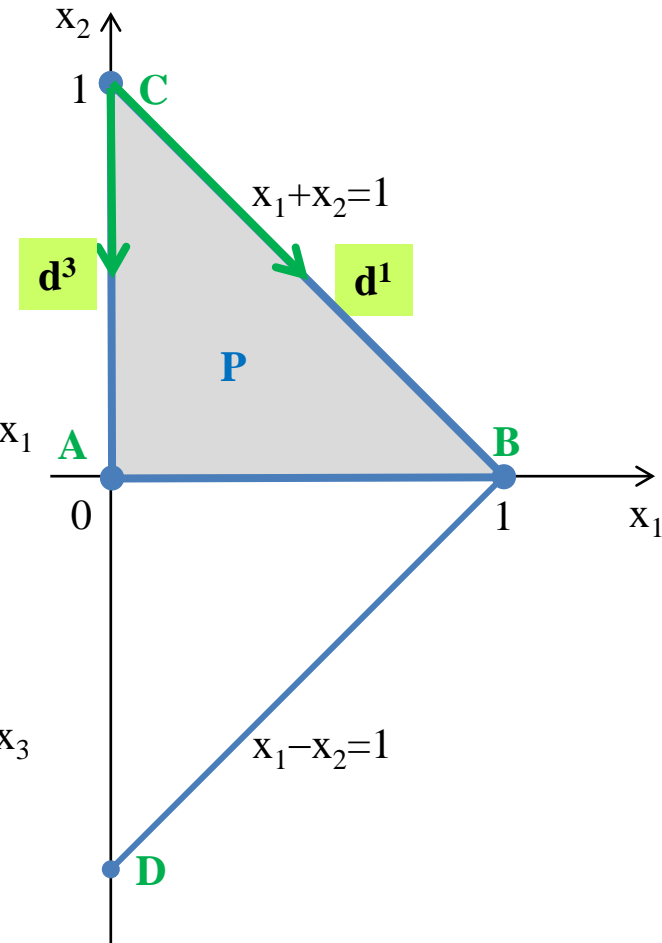
$$d_B = -B^{-1}A_{.,1} = -\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$$

$$d^1 = (1 \ -1 \ 0 \ -2) \rightarrow \text{admissible}$$

- **Direction de base d^3** correspondant à la variable hors base x_3

$$d_B = -B^{-1}A_{.,3} = -\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

$$d^3 = (0 \ -1 \ 1 \ -1) \rightarrow \text{admissible}$$



1.2.2 Exemple

Recherche des directions de base

Examen de directions de base de P

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}$$

- **Base admissible (x_1, x_4)** : $x = (1 \ 0 \ 0 \ 0) \rightarrow$ **point B**
 base dégénérée

$$B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$$

- **Direction de base d^2** correspondant à la variable hors base x_2

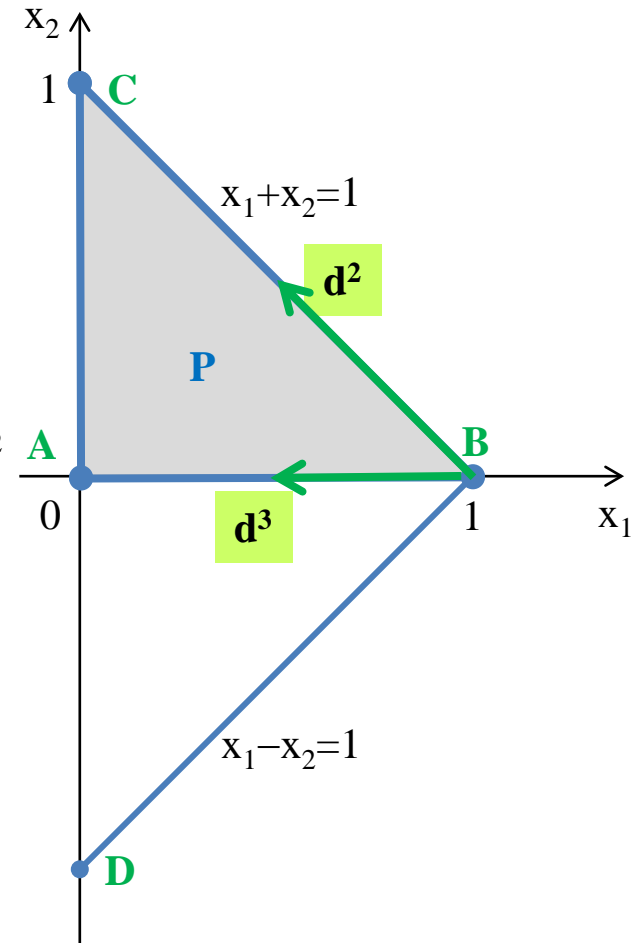
$$d_B = -B^{-1}A_{.,2} = -\begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

$$d^2 = (-1 \ 1 \ 0 \ 2) \rightarrow \text{admissible}$$

- **Direction de base d^3** correspondant à la variable hors base x_3

$$d_B = -B^{-1}A_{.,3} = -\begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$d^3 = (-1 \ 0 \ 1 \ 1) \rightarrow \text{admissible}$$



1.2.2 Exemple

Recherche des directions de base

Examen de directions de base de P

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}$$

- **Base admissible (x_1, x_2)** : $x = (1 \ 0 \ 0 \ 0) \rightarrow$ **point B**
 base dégénérée

$$B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, B^{-1} = \begin{pmatrix} 0,5 & 0,5 \\ 0,5 & -0,5 \end{pmatrix}$$

- **Direction de base d^3** correspondant à la variable hors base x_3

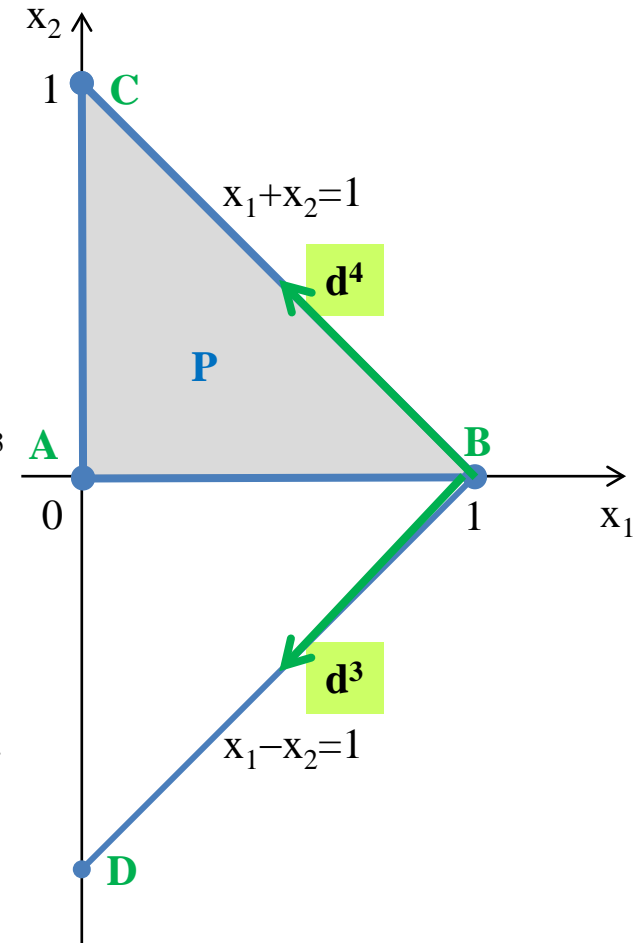
$$d_B = -B^{-1}A_{.,3} = -\begin{pmatrix} 0,5 & 0,5 \\ 0,5 & -0,5 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -0,5 \\ -0,5 \end{pmatrix}$$

$$d^3 = (-0,5 \ -0,5 \ 1 \ 0) \rightarrow \text{non admissible (base dégénérée)}$$

- **Direction de base d^4** correspondant à la variable hors base x_4

$$d_B = -B^{-1}A_{.,4} = -\begin{pmatrix} 0,5 & 0,5 \\ 0,5 & -0,5 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -0,5 \\ 0,5 \end{pmatrix}$$

$$d^4 = (-0,5 \ 0,5 \ 0 \ 1) \rightarrow \text{admissible}$$



1.2.3 Réduction

- ☐ Principe
- ☐ Méthode générale
- ☐ Réduction avec noyau
- ☐ Choix des matrices Y et Z
- ☐ Interprétation géométrique

1.2.3 Réduction

Problème sous contraintes linéaires

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } Ax = b, \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, \quad A \text{ de rang plein } r=m \leq n$$

- On cherche un déplacement p à partir d'un point initial x_0 .
Le point initial x_0 n'est pas forcément admissible : $Ax_0 = b_0$.
- Le nouveau point x doit être admissible et meilleur que x_0 (diminution du critère).

$$x_0 \rightarrow x = x_0 + p \text{ avec } \begin{cases} Ax = b & \rightarrow \text{admissible} \\ f(x) < f(x_0) & \rightarrow \text{amélioration} \end{cases}$$

Principe de réduction

On utilise les m contraintes pour réduire le problème à $n-m$ variables.

Le déplacement p est décomposé en 2 termes : $p = p_{\text{libre}} + p_{\text{lié}}$, $p_{\text{libre}} \in \mathbb{R}^n$, $p_{\text{lié}} \in \mathbb{R}^n$

- p_{libre} dépend de $n-m$ variables libres (ou indépendantes) \rightarrow pour minimiser le critère f
- $p_{\text{lié}}$ dépend de m variables liées (ou dépendantes) \rightarrow pour restaurer l'admissibilité

- $p_{\text{lié}}$ est calculé à partir des contraintes

$$Ax = b \Rightarrow A(x_0 + p) = b \Rightarrow A(p_{\text{libre}} + p_{\text{lié}}) = b - b_0 \Rightarrow Ap_{\text{lié}} = b - b_0 - Ap_{\text{libre}}$$

\rightarrow Système non singulier (A de rang plein)

\rightarrow Le problème d'optimisation est réduit à p_{libre} ($n-m$ variables)

1.2.3 Méthode générale

Décomposition du déplacement

- On choisit une base de \mathbb{R}^n formée de n vecteurs indépendants : $(y_1, \dots, y_m, z_1, \dots, z_{n-m})$
 Le déplacement p s'écrit comme une combinaison linéaire des vecteurs y_i et z_i :

$$p = \sum_{i=1}^m a_i y_i + \sum_{i=1}^{n-m} b_i z_i \Leftrightarrow \boxed{p = Yp_Y + Zp_Z}$$

avec matrice $Y = (y_1, \dots, y_m) \in \mathbb{R}^{n \times m}$ = composantes des m vecteurs y_1, \dots, y_m
 matrice $Z = (z_1, \dots, z_{n-m}) \in \mathbb{R}^{n \times (n-m)}$ = composantes des $n-m$ vecteurs z_1, \dots, z_{n-m}
 vecteur $p_Y = (a_1, \dots, a_m) \in \mathbb{R}^m$ = coefficients des m vecteurs y_1, \dots, y_m
 vecteur $p_Z = (b_1, \dots, b_{n-m}) \in \mathbb{R}^{n-m}$ = coefficients des $n-m$ vecteurs z_1, \dots, z_{n-m}

- Les composantes liées et libres du déplacement $p = p_{\text{lié}} + p_{\text{libre}}$ sont définies par :

$$\mathbf{p}_{\text{lié}} = \mathbf{Y} \mathbf{p}_Y \in \mathbb{R}^n \quad \rightarrow m \text{ variables liées } (p_Y)$$

$$\mathbf{p}_{\text{libre}} = \mathbf{Z} \mathbf{p}_Z \in \mathbb{R}^n \quad \rightarrow n-m \text{ variables libres } (p_Z)$$

- Le déplacement doit être admissible

$$\begin{aligned} A(x_0 + p) = b &\Rightarrow A p_{\text{lié}} = b - b_0 - A p_{\text{libre}} \\ &\Rightarrow A Y p_Y = b - b_0 - A Z p_Z \\ &\Rightarrow p_Y = (A Y)^{-1} (b - b_0 - A Z p_Z) \quad \text{si la matrice } A Y \text{ est inversible} \end{aligned}$$

1.2.3 Méthode générale

Problème équivalent sans contrainte

- Déplacement total

$$p = Yp_Y + Zp_Z \quad \text{avec} \quad p_Y = (AY)^{-1}(b - b_0 - AZp_Z) \quad \text{si la matrice } AY \text{ est inversible}$$

$$\Rightarrow p = Y(A Y)^{-1}(b - b_0) + (I - Y(A Y)^{-1} A) Z p_Z \quad \rightarrow \text{réduction à } n-m \text{ variables } p_Z$$

- **Coût réduit**

$$f(x) = f(x_0 + p) = f\left(x_0 + Y(A Y)^{-1}(b - b_0) + (I - Y(A Y)^{-1} A) Z p_Z\right)$$

$$\underset{\text{noté}}{=} \varphi(p_Z) \quad \rightarrow \text{coût réduit } \varphi = \text{fonction de } n-m \text{ variables}$$

$\min_{p \in \mathbb{R}^n} f(x_0 + p) \text{ sous } A(x_0 + p) = b$	\longrightarrow	$n \text{ variables } p / m \text{ contraintes}$
$\Leftrightarrow \min_{p_Z \in \mathbb{R}^{n-m}} \varphi(p_Z)$	\longrightarrow	$n-m \text{ variables } p_Z / 0 \text{ contrainte}$

Choix des matrices Y et Z

- Réduction avec noyau \rightarrow respect des contraintes avec p_Y , minimisation avec p_Z
- Matrices orthogonales \rightarrow meilleur conditionnement

1.2.3 Réduction avec noyau

Base du noyau

- On choisit pour les $n-m$ vecteurs (z_1, \dots, z_{n-m}) une **base de l'espace nul de A** : $Az_i=0$
 + m vecteurs (y_1, \dots, y_m) pour former une base de \mathbb{R}^n
 $\Rightarrow \mathbf{AZ} = \mathbf{0}$

- La matrice $(Y \ Z) \in \mathbb{R}^{n \times n}$ est de rang plein (base de \mathbb{R}^n)
 $\Rightarrow A(Y \ Z) = (AY \ 0)$ de rang plein
 $\Rightarrow \mathbf{AY}$ **matrice inversible** de $\mathbb{R}^{m \times m}$

- Le déplacement lié se simplifie

$$p_{\text{lié}} = Yp_Y = Y(AY)^{-1}(b - b_0 - AZp_Z) = Y(AY)^{-1}(b - b_0)$$

$\rightarrow p_{\text{lié}}$ est constant et indépendant de $p_{\text{libre}} = Zp_Z$

$$p = Yp_Y + Zp_Z = Y(AY)^{-1}(b - b_0) + Zp_Z$$

Problème réduit

$$\min_{p \in \mathbb{R}^n} f(x_0 + p) \text{ sous } A(x_0 + p) = b \Leftrightarrow \min_{p_Z \in \mathbb{R}^{n-m}} \varphi(p_Z) = f\left(x_0 + Y(AY)^{-1}(b - b_0) + Zp_Z\right)$$

1.2.3 Choix des matrices Y et Z

A partir d'une base de A

- On choisit une **base B** $\in \mathbb{R}^{m \times m}$ de la matrice A (= **m colonnes indépendantes de A**)

$$AE = \begin{pmatrix} m & n-m \\ B & N \end{pmatrix} \quad E^T p = \begin{pmatrix} p_B \\ p_N \end{pmatrix} \begin{matrix} \rightarrow m \\ \rightarrow n-m \end{matrix} \quad (E = \text{matrice de permutation de colonnes de A})$$

- Le déplacement p est décomposé en $p = p_{\text{libre}} + p_{\text{lié}}$ avec $p_{\text{lié}} \in \mathbb{R}^n$, $p_{\text{libre}} \in \mathbb{R}^n$, définis par :

$$E^T p_{\text{lié}} = \begin{pmatrix} p_B \\ 0 \end{pmatrix} \quad E^T p_{\text{libre}} = \begin{pmatrix} 0 \\ p_N \end{pmatrix} \quad Ap_{\text{lié}} = b - b_0 - Ap_{\text{libre}} \Rightarrow \boxed{p_B = B^{-1}(b - b_0) - B^{-1}Np_N}$$

- La décomposition correspondante dans \mathbb{R}^n est $p = Yp_Y + Zp_Z$ avec

$$Y = \begin{pmatrix} m & n-m \\ B^{-1} & -B^{-1}N \end{pmatrix} \begin{matrix} m \\ n-m \end{matrix} \quad Z = \begin{pmatrix} 0 \\ I \end{pmatrix}_{n-m} \quad \Rightarrow p = Yp_Y + Zp_Z = \begin{pmatrix} p_B \\ p_N \end{pmatrix}$$

$$p_Y = b - b_0 \quad p_Z = p_N \quad \rightarrow \text{décomposition directe selon les composantes de p}$$

Problème réduit

$$\min_{p \in \mathbb{R}^n} f(x_0 + p) \text{ sous } A(x_0 + p) = b \Leftrightarrow \min_{p_N \in \mathbb{R}^{n-m}} \varphi(p_N) = f(x_{0B} + B^{-1}(b - b_0 - Np_N), x_{0N} + p_N)$$

1.2.3 Choix des matrices Y et Z

A partir de matrices orthogonales

- La matrice AY doit être inversée

$$p = Yp_Y + Zp_Z = Y(AY)^{-1}(b - b_0) + Zp_Z$$

→ Il faut choisir la base (Y Z) pour obtenir le meilleur conditionnement possible

- **Factorisation QR** de A

$A^T E = QR$ avec Q orthogonale ($QQ^T = I$) → méthode de Householder
 R triangulaire

$$A^T E = \begin{pmatrix} m & n-m \\ Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} m \\ R \\ 0 \end{pmatrix}_{n-m} \rightarrow \begin{cases} Y = Q_1 \\ Z = Q_2 \end{cases}$$

- Conditionnement de AY

$$A^T E = Q_1 R \Rightarrow A = E R^T Q_1^T \Rightarrow AY = E R^T Q_1^T Q_1 = E R^T$$

→ même conditionnement que R

→ même conditionnement que A (car $QQ^T = I \rightarrow$ conditionnement = 1)
 = **conditionnement minimal** possible à partir de A

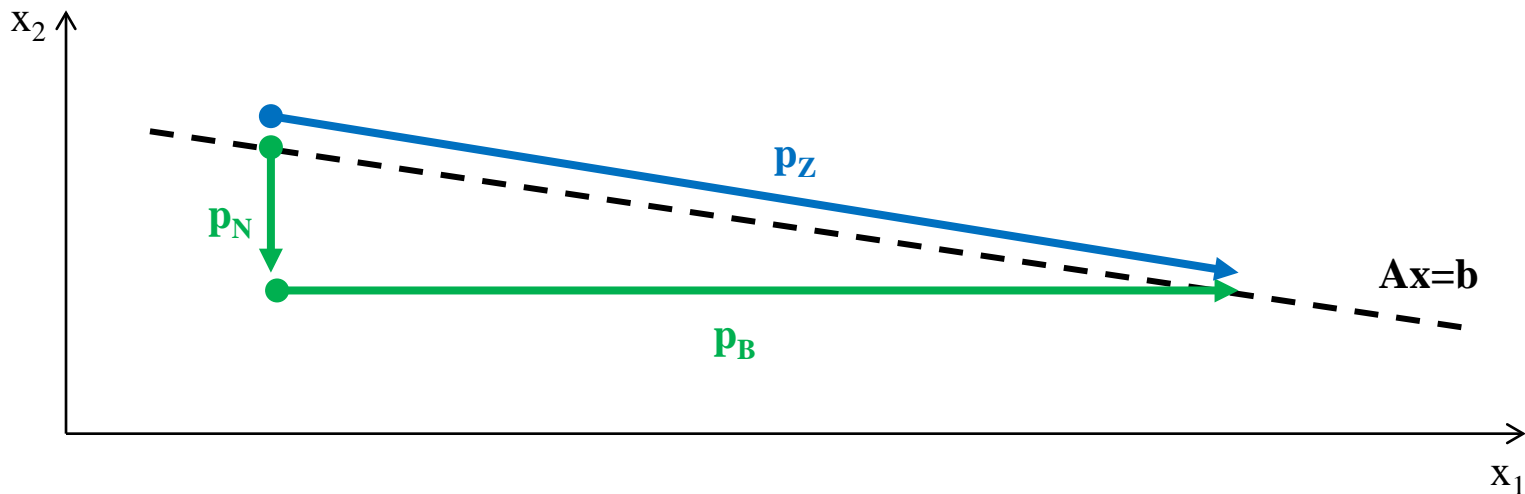
1.2.3 Interprétation géométrique

A partir d'une base de A

- $p_N = n-m$ composantes de x → minimisation de f
- $p_B = m$ composantes de x → restauration de $Ax=b$
 $p_B \neq 0$ car p_N ne tient pas compte les contraintes → mauvais conditionnement si $p_B \gg p_N$

A partir de matrices orthogonales

- Zp_Z = déplacement dans le noyau x → minimisation de f
- Yp_Y = déplacement orthogonal à Zp_Z → restauration de $Ax=b$
 $p_Y=0$ si x_0 est admissible car p_Z conserve les contraintes → meilleur conditionnement



1.2.4 Projection

- ❑ Projection orthogonale sur un hyperplan
- ❑ Projection sur le noyau

1.2.4 Projection

Projection orthogonale sur un hyperplan

La projection orthogonale de $x_0 \in \mathbb{R}^n$ sur l'hyperplan d'équation $Ax=b$ est le point x solution de

$$\boxed{\min_{x \in \mathbb{R}^n} \|x - x_0\| \text{ sous } Ax = b} \rightarrow \text{point } x_P \text{ de l'hyperplan le plus proche de } x_0$$

- Problème quadratique équivalent

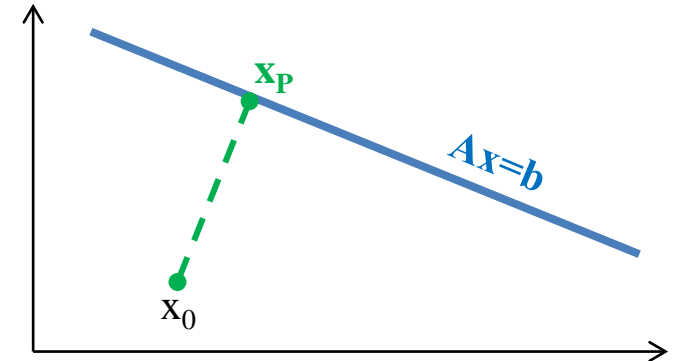
$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|x - x_0\|^2 = \frac{1}{2} (x - x_0)^T (x - x_0) \text{ sous } Ax = b$$

- Lagrangien : $L(x, \lambda) = \frac{1}{2} (x - x_0)^T (x - x_0) + \lambda^T (b - Ax)$

- Condition d'ordre 1

$$\begin{cases} x - x_0 - A^T \lambda = 0 \\ Ax = b \end{cases} \Rightarrow \begin{cases} Ax - AA^T \lambda = Ax_0 \\ Ax = b \end{cases} \Rightarrow \begin{cases} \lambda = (AA^T)^{-1} (b - Ax_0) \\ x = x_0 + A^T (AA^T)^{-1} (b - Ax_0) \end{cases}$$

- Solution : $x_P = \left(I - A^T (AA^T)^{-1} A \right) x_0 + A^T (AA^T)^{-1} b$



- Projection de x_0 sur le noyau de $A : Ax=0 \rightarrow x_P = \left(I - A^T (AA^T)^{-1} A \right) x_0$
 \rightarrow **matrice de projection** : $P = I - A^T (AA^T)^{-1} A$

Sommaire

- 1. Bases théoriques**
 - 1.1 Définitions
 - 1.2 Contraintes linéaires
 - 1.3 Contraintes non linéaires**
 - 1.3.1 Direction admissible
 - 1.3.2 Déplacement admissible
 - 1.4 Conditions d'optimalité
- 2. Optimisation sans contraintes
- 3. Optimisation avec contraintes

1.3.1 Direction admissible

- ☐ Indépendance linéaire
- ☐ Direction admissible à la limite
- ☐ Cône des directions
- ☐ Qualification des contraintes

1.3.1 Indépendance linéaire

Contraintes linéaires

Pour des contraintes linéaires $Ax=b$, $A \in \mathbb{R}^{m \times n}$, si A est de rang déficient : $\text{rang}(A) = r < m$, on peut toujours extraire de A une sous-matrice $\tilde{A} \in \mathbb{R}^{r \times n}$ de rang plein : $\text{rang}(\tilde{A}) = r$, telle que :
 $\tilde{A}x = \tilde{b} \Leftrightarrow Ax = b \rightarrow$ élimination des contraintes redondantes (cf §1.2.1)

Contraintes non linéaires

Pour des contraintes non linéaires, on considère un modèle linéaire local.

- x_0 point admissible : $\begin{cases} c_E(x_0) = 0 \\ c_I(x_0) \leq 0 \end{cases} \Leftrightarrow c(x_0) = 0$ (contraintes actives en x_0)
- Contraintes actives linéarisées : $\hat{c}_0(x) = c(x_0) + \nabla c(x_0)^T (x - x_0)$ avec $c(x_0) = 0$
 $\hat{c}_0(x) = 0 \Leftrightarrow \nabla c(x_0)^T x = \nabla c(x_0)^T x_0 \Leftrightarrow Ax = b$

On se ramène au cas de contraintes linéaires avec $A = \nabla c(x_0)^T$ (gradient des contraintes actives)

Condition d'indépendance linéaire

Les contraintes sont dites **linéairement indépendantes** en x_0 si les gradients des contraintes actives sont linéairement indépendants en x_0 .
 \Leftrightarrow La matrice jacobienne des contraintes actives $J(x_0) = \nabla c(x_0)$ est de rang plein.

1.3.1 Exemple

Indépendance linéaire

- 1 contrainte égalité + 1 contrainte inégalité dans \mathbb{R}^2

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 \quad \begin{cases} c_1(\mathbf{x}) = x_2 - x_1^2 = 0 \\ c_2(\mathbf{x}) = x_1^2 + (x_2 - 1)^2 - 1 \leq 0 \end{cases}$$

- En $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

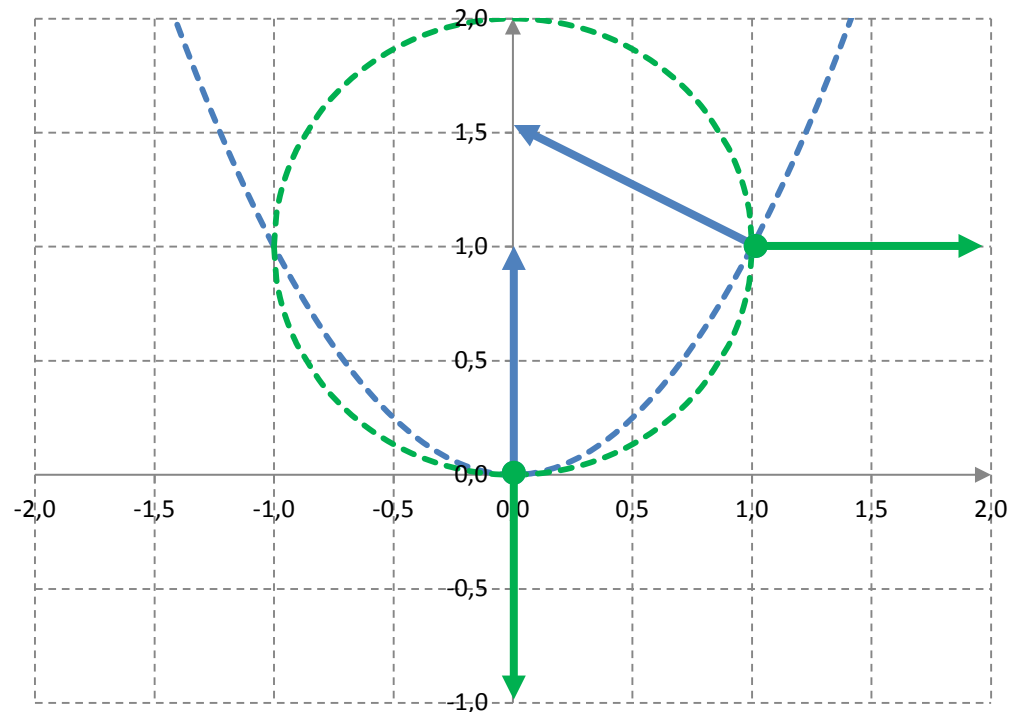
$$\nabla c_1 = \begin{pmatrix} -2 \\ 1 \end{pmatrix} \quad \nabla c_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

→ **linéairement indépendants**

- En $\mathbf{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$\nabla c_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \nabla c_2 = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

→ **linéairement dépendants**



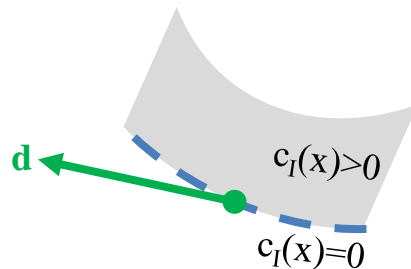
1.3.1 Direction admissible

Définition générale

- x point admissible : $\begin{cases} c_E(x_0) = 0 \\ c_I(x_0) \leq 0 \end{cases} \Leftrightarrow c(x_0) = 0$ (contraintes actives en x_0)
- **d direction admissible à partir de x** $\Leftrightarrow \exists \eta > 0 / \forall s, 0 < s < \eta, x + sd$ admissible
 On peut se déplacer sur un segment de longueur ε suivant d à partir de x en restant admissible.

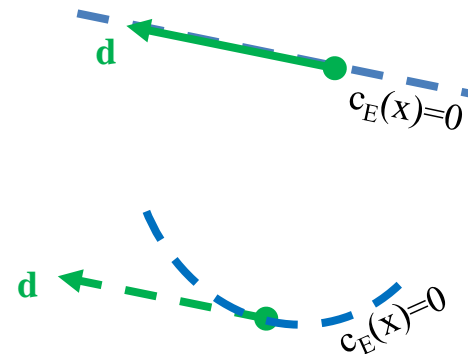
Applicabilité

- Applicable aux contraintes inégalité et aux contraintes égalité linéaires



- Inapplicable aux contraintes égalité linéaires

→ Définition à partir de suites de points admissibles



1.3.1 Direction admissible à la limite

Suite de points admissibles

x point admissible

Définition : $(x_k)_{k \in \mathbb{N}}$ **suite admissible** en $x \Leftrightarrow \begin{cases} \forall k, x_k \neq x \\ \lim_{k \rightarrow \infty} x_k = x \\ \exists k_0 / \forall k \geq k_0, x_k \text{ admissible} \end{cases}$

Direction admissible à la limite

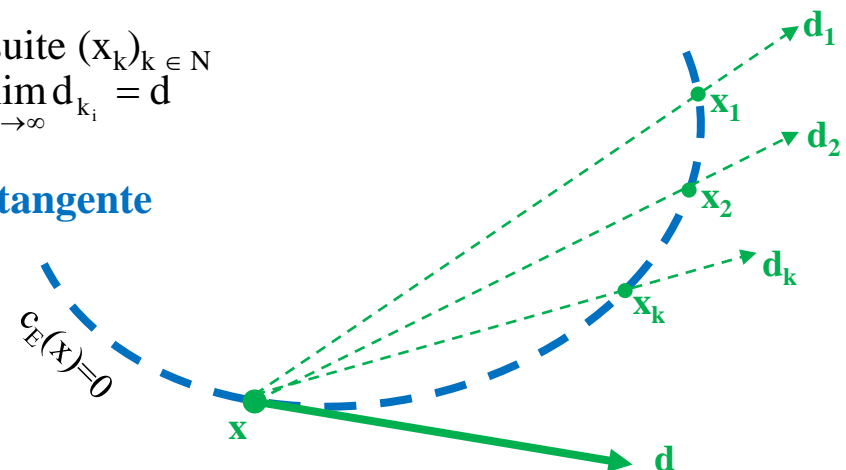
- On considère la **suite des directions d_k reliant x_k à x** : $d_k = \frac{x_k - x}{\|x_k - x\|}$

- Définition

d direction admissible à la limite en x pour la suite $(x_k)_{k \in \mathbb{N}}$

\Leftrightarrow Il existe une sous-suite $(d_{k_i})_{i \in \mathbb{N}}$ telle que : $\lim_{i \rightarrow \infty} d_{k_i} = d$

- Direction admissible à la limite = direction tangente**



1.3.1 Cône des directions

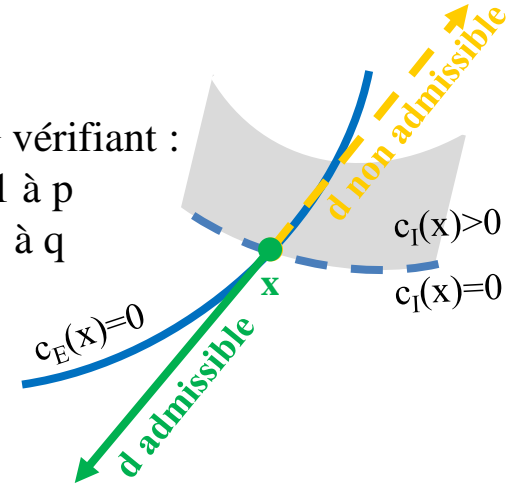
Définition

x point admissible

Le **cône des directions $D(x)$ en x** est l'ensemble des directions $d \in \mathbb{R}^n$ vérifiant :

- $\nabla c_{Ej}(x)^T d = 0$ pour toutes les contraintes égalité $c_{Ej}(x) = 0, j=1 \text{ à } p$
- $\nabla c_{Ij}(x)^T d \leq 0$ pour les contraintes inégalité actives : $c_{Ij}(x) = 0, j=1 \text{ à } q$

$d \in D(x)$ → direction **tangente** aux contraintes égalité
 → direction **intérieure** aux contraintes inégalité actives



Propriété

Toute direction admissible à la limite en x appartient au cône des directions en x

Preuve :

$$(x_k) \text{ suite admissible de limite } x \Rightarrow \begin{cases} c_E(x_k) = 0 \\ c_I(x_k) \leq 0 \end{cases} \rightarrow \text{directions } d_k = \frac{x_k - x}{\|x_k - x\|}$$

$$c \text{ contrainte active en } x : c(x) = 0 \quad c(x_k) = c(x) + \nabla c(x)^T (x_k - x) + o(\|x_k - x\|)$$

$$\nabla c(x)^T d_k = \frac{c(x_k) - c(x)}{\|x_k - x\|} - \frac{o(\|x_k - x\|)}{\|x_k - x\|} \Rightarrow \nabla c(x)^T d = \lim_{k \rightarrow \infty} \frac{c(x_k)}{\|x_k - x\|} \rightarrow \begin{cases} = 0 & (\text{égalité}) \\ \leq 0 & (\text{inégalité}) \end{cases}$$

1.3.1 Qualification

Caractérisation des directions admissibles

- Le cône des directions $D(x)$ au point x admissible est simple à manipuler en pratique :

$$d \in D(x) \Leftrightarrow \begin{cases} \nabla c_E(x)^T d = 0 & \rightarrow \text{pour toutes les contraintes égalité} \\ \nabla c_I(x)^T d \leq 0 & \rightarrow \text{pour les contraintes inégalité actives en } x \end{cases}$$

- Toutes les directions admissibles à la limite en x appartiennent à $D(x)$, mais $D(x)$ peut contenir également des directions non admissibles.
 $\rightarrow D(x)$ ne caractérise pas les directions admissibles.

Qualification des contraintes

Les contraintes vérifient la **condition de qualification** au point admissible x si toute direction du cône $D(x)$ est admissible à la limite.

\rightarrow Condition très importante dans les algorithmes

Conditions suffisantes de qualification des contraintes

- Contraintes linéaires : $Ax=b$
ou
- Contraintes **linéairement indépendantes en x** : $\nabla c(x)$ de rang plein
 \rightarrow réalisable simplement en pratique par extraction d'une sous-matrice de rang plein

1.3.2 Déplacement admissible

- ☐ Principes
- ☐ Elimination directe
- ☐ Réduction généralisée
- ☐ Restauration

1.3.2 Déplacement admissible

Problème sous contraintes non linéaires

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$$

- On cherche à construire un déplacement p admissible et améliorant à partir d'un point initial x_0 .
On se ramène à un problème avec contraintes égalité (contraintes actives en x_0).

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow \quad m \text{ contraintes actives en } x_0$$

- Les n composantes du déplacement p doivent vérifier :
$$\begin{cases} c(x_0 + p) = 0 \\ f(x_0 + p) < f(x_0) \end{cases}$$

Méthodes possibles

- Elimination directe
On exprime m variables à partir des $n-m$ autres à partir des contraintes.
On substitue dans l'expression de f \rightarrow problème sans contraintes
- Réduction généralisée
On linéarise les contraintes en x_0 .
On applique la méthode de réduction des contraintes linéaires (matrices Y et Z).
On corrige le déplacement pour prendre en compte les non-linéarités.

1.3.2 Elimination directe

Principe

$\min_{x \in \mathbb{R}^n} f(x)$ sous $c(x) = 0 \rightarrow m$ contraintes actives

- Les contraintes sont de la forme : $c(x) = c(x_{\text{lié}}, x_{\text{libre}})$, $x_{\text{lié}} \in \mathbb{R}^m$, $x_{\text{libre}} \in \mathbb{R}^{n-m}$
- Si l'on sait résoudre : $c(x_{\text{lié}}, x_{\text{libre}}) = 0 \Leftrightarrow x_{\text{lié}} = \psi(x_{\text{libre}})$

le problème devient : $\min_{x_{\text{libre}} \in \mathbb{R}^{n-m}} \varphi(x_{\text{libre}})$ avec $\varphi(x_{\text{libre}}) \stackrel{\text{def}}{=} f(x_{\text{lié}}, x_{\text{libre}}) = f(\psi(x_{\text{libre}}), x_{\text{libre}})$

\rightarrow problème de dimension $n-m$, sans contrainte

Difficultés

- Il faut faire attention au **domaine de définition** des variables (contraintes implicites)
 \rightarrow voir exemples
- Il faut disposer de l'expression analytique des fonctions (rarement réalisé en pratique)

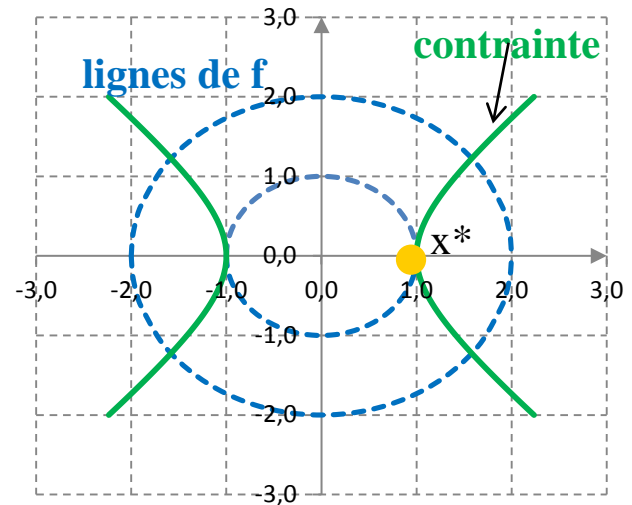
1.3.2 Exemples

Elimination directe

- Exemple 1 : $\min_{x_1, x_2} x_1^2 + x_2^2$ sous $x_1^2 - x_2^2 = 1$

Elimination de x_1 : $x_1^2 - x_2^2 = 1 \Rightarrow x_1^2 = 1 + x_2^2$
 $\rightarrow \min_{x_2} 1 + 2x_2^2 \Rightarrow x_2 = 0$

Solution correcte : $\begin{cases} x_1 = 1 \\ x_2 = 0 \end{cases}$



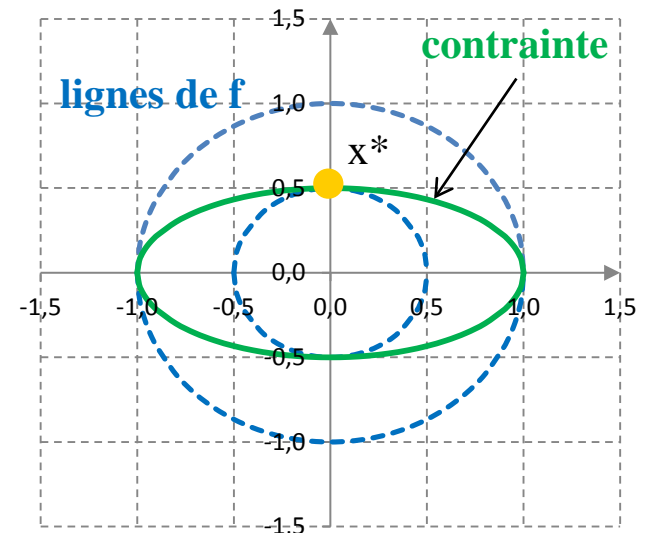
- Exemple 2 : $\min_{x_1, x_2} x_1^2 + x_2^2$ sous $x_1^2 + 4x_2^2 = 1$

Elimination de x_1 : $x_1^2 + 4x_2^2 = 1 \Rightarrow x_1^2 = 1 - 4x_2^2$
 $\rightarrow \min_{x_2} 1 - 3x_2^2 \Rightarrow x_2 = \pm\infty$

Solution incorrecte

Contrainte implicite : $x_1^2 \geq 0 \Rightarrow 1 - 4x_2^2 \geq 0$
 $\Rightarrow -\frac{1}{2} \leq x_2 \leq \frac{1}{2}$

\rightarrow à prendre en compte explicitement dans la résolution



1.3.2 Réduction généralisée

Principe

$\min_{x \in \mathbb{R}^n} f(x)$ sous $c(x) = 0 \rightarrow m$ contraintes actives

On construit le déplacement p à partir du point initial x_0 en 2 étapes : $p = p_1 + p_2$

- **Etape de linéarisation + réduction**

Le déplacement p_1 améliore le critère en supposant un modèle linéaire des contraintes en x_0 .

→ linéarisation des contraintes en x_0

→ application de la méthode de réduction de contraintes linéaires

On obtient un nouveau point : $x_1 = x_0 + p_1$

- **Etape de restauration**

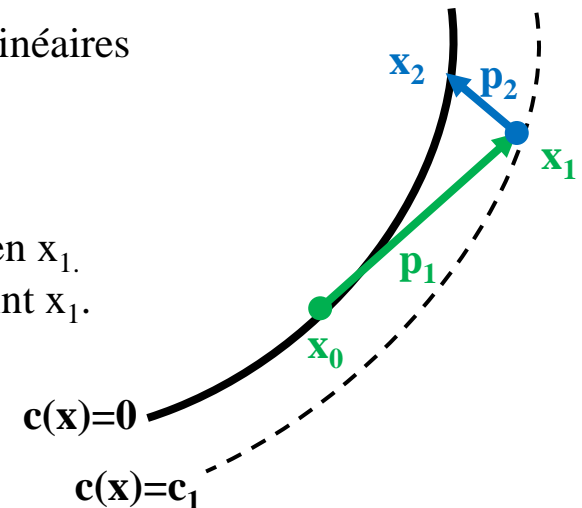
Les contraintes actives (non linéaires) ne sont pas respectées en x_1 .

Le déplacement p_2 restaure un point admissible à partir du point x_1 .

→ linéarisation des contraintes en x_1

→ résolution d'un système sous-déterminé

On obtient un nouveau point : $x_2 = x_1 + p_2$



- Le point x_2 doit être : - admissible pour l'ensemble des contraintes (actives et inactives en x_0)
 - meilleur que x_0 ($f(x_2) < f(x_0)$)

1.3.2 Réduction généralisée

Etape de linéarisation + réduction

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \rightarrow m \text{ contraintes actives}$$

- On linéarise les contraintes au point initial x_0 : $\hat{c}_0(x) = c(x_0) + \nabla c(x_0)^T (x - x_0)$

- Le déplacement p est admissible pour les contraintes linéaires si :

$$\hat{c}_0(x_0 + p) = 0 \Leftrightarrow c(x_0) + \nabla c(x_0)^T p = 0 \Leftrightarrow A_0 p = b_0 \text{ avec } \begin{cases} A_0 = \nabla c(x_0)^T \\ b_0 = -c(x_0) \end{cases}$$

- On applique la méthode de réduction de contraintes linéaires.

Le déplacement est décomposé en $p = Yp_Y + Zp_Z$ avec Z base du noyau de A_0

$$\min_{p \in \mathbb{R}^n} f(x_0 + p) \text{ sous } A_0 p = b_0 \Leftrightarrow \min_{p_Z \in \mathbb{R}^{n-m}} \varphi(p_Z) = f(x_0 + Y(A_0 Y)^{-1} b_0 + Zp_Z)$$

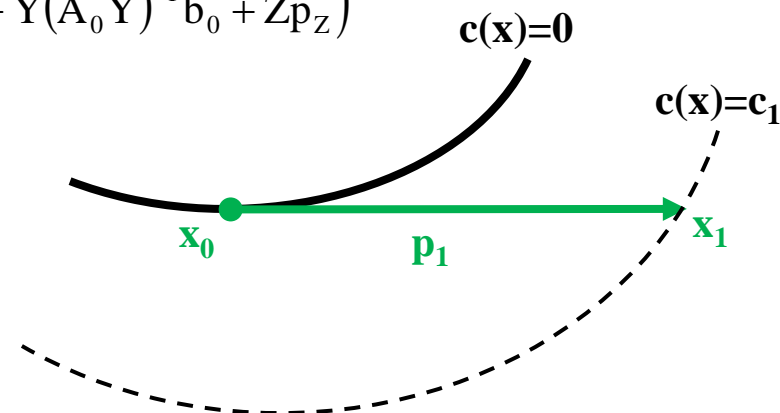
→ problème à $n-m$ variables sans contraintes

→ **déplacement p_1**

- Le nouveau point $x_1 = x_0 + p_1$

- est meilleur que x_0 : $f(x_1) < f(x_0)$

- ne vérifie pas les contraintes : $c(x_1) = c_1 \neq 0$



1.3.2 Réduction généralisée

Etape de restauration

- Les contraintes actives ne sont pas vérifiées en x_1 : $c(x_1) = c_1 \neq 0$
 On cherche un déplacement p_2 à partir de x_1 tel que : $c(x_1 + p_2) = 0$
 → système non linéaire sous-déterminé de m équations à n inconnues
- On linéarise les contraintes au point x_1 : $\hat{c}_1(x) = c(x_1) + \nabla c(x_1)^T (x - x_1)$
 On obtient un **système linéaire sous-déterminé** de m équations à n inconnues :

$$\hat{c}_1(x_1 + p) = 0 \Leftrightarrow c(x_1) + \nabla c(x_1)^T p = 0 \Leftrightarrow A_1 p = b_1 \quad \text{avec} \quad \begin{cases} A_1 = \nabla c(x_1)^T \\ b_1 = -c(x_1) \end{cases}$$

Résolution du système

- Il faut recalculer les gradients des contraintes en x_1 → très coûteux ($m \times n$ appels fonction)
 On fait l'approximation que : $\nabla c(x_1) \approx \nabla c(x_0)$
 → approximation correcte si le déplacement p_1 est petit
 ou si les contraintes sont peu non linéaires
- Le système sous-déterminé admet une infinité de solutions ($n - m$ variables libres)
 Choix possibles :
 - solution de norme minimale (projection sur les contraintes)
 - solution de base (pour ne pas dégrader la minimisation due à p_1)

1.3.2 Restauration

Résolution du système

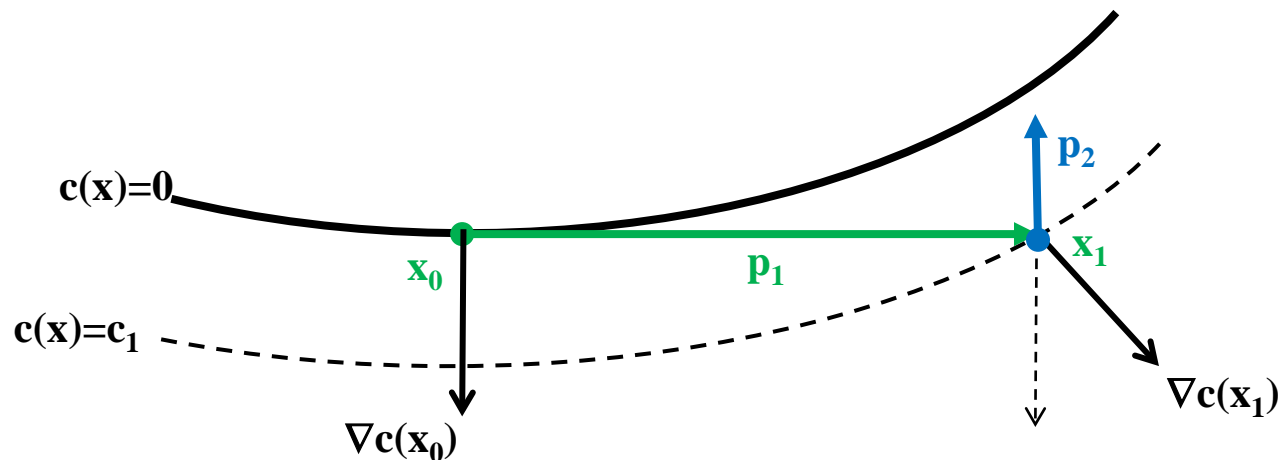
Le déplacement p_2 doit vérifier : $A_1 p = b_1$ avec $\begin{cases} A_1 = \nabla c(x_1)^T \approx \nabla c(x_0)^T = A_0 \\ b_1 = -c(x_1) = -c_1 \end{cases}$

- **Solution de norme minimale** → projection sur l'hyperplan tangent aux contraintes actives (cf §1.2.4)

$$\min_{p \in \mathbb{R}^n} \|p\| \text{ sous } A_1 p = b_1 \quad \rightarrow \quad p_2 = A_1^T (A_1 A_1^T)^{-1} b_1$$

- **Solution de base** → pour ne pas dégrader la minimisation réalisée par p_Z (cf §1.2.3)

$$A_1(Yp_Y + Zp_Z) = b_1 \Rightarrow p_Y = (A_1 Y)^{-1} b_1 \rightarrow p_2 = Y(A_1 Y)^{-1} b_1$$



1.3.2 Restauration

Itérations

- La résolution est basée sur une linéarisation du système en x_1 .

$$A_1 p = b_1 \quad \text{avec} \quad \begin{cases} A_1 = \nabla c(x_1)^T \approx \nabla c(x_0)^T = A_0 \\ b_1 = -c(x_1) = -c_1 \end{cases} \rightarrow \text{déplacement } p_2$$

- Le point $x_2 = x_1 + p_2$ ne vérifie pas forcément les contraintes : $c(x_2) = c_2 \neq 0$

Il faut alors réitérer la résolution à partir de x_2 .

- On cherche un déplacement p_3 à partir de x_2 tel que : $c(x_2 + p_3) = 0$

$$A_2 p = b_2 \quad \text{avec} \quad \begin{cases} A_2 = \nabla c(x_2)^T \approx \nabla c(x_0)^T = A_0 \\ b_2 = -c(x_2) = -c_2 \end{cases} \rightarrow \text{déplacement } p_3$$

- Si l'on n'obtient pas de point admissible après un nombre donné d'itérations,
il faut réduire le déplacement p_1 pour rendre l'approximation linéaire suffisamment correcte.

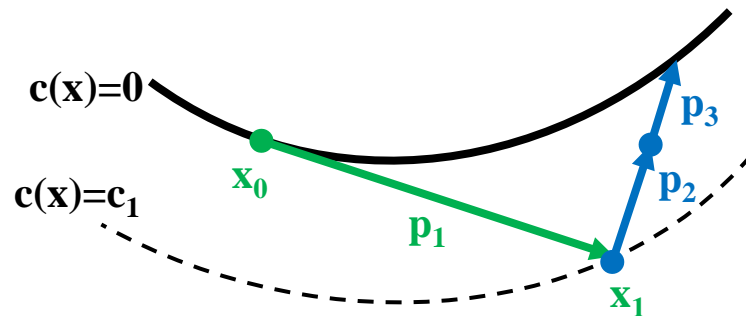
- Si l'on obtient un point admissible x_2 , il faut encore vérifier que :
 - les contraintes qui étaient inactives en x_0 sont respectées en x_2
 - le point x_2 est meilleur que le point initial x_0 (car la restauration peut dégrader le critère).

Sinon il faut réduire le déplacement p_1 .

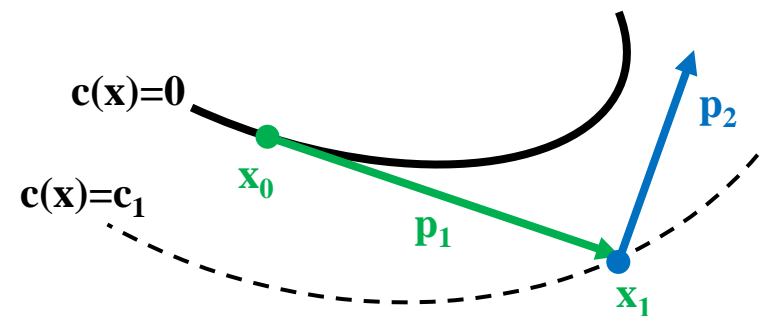
1.3.2 Restauration

Illustrations

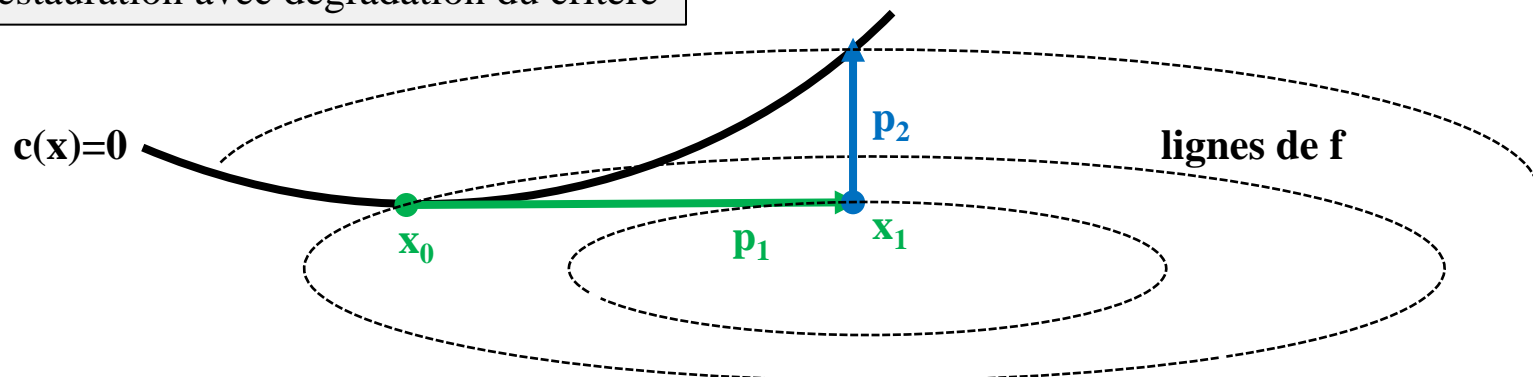
Restauration en plusieurs itérations : p_2, p_3



Restauration infructueuse (non linéarité)



Restauration avec dégradation du critère



1.3.2 Direction d'ordre 2

Effet Maratos

La restauration après le déplacement p_1 peut dégrader systématiquement le critère.

Il faut alors réduire fortement le pas p_1 pour progresser.

Ceci peut bloquer un algorithme basé sur une recherche linéaire suivant p_1 (effet Maratos).

Correction d'ordre 2

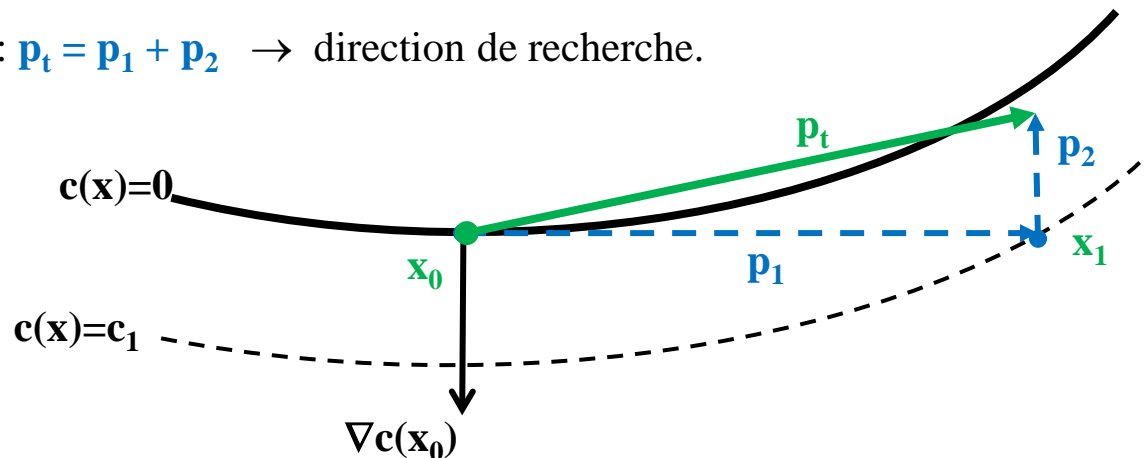
- On corrige la direction de déplacement pour prendre en compte la non-linéarité des contraintes.

p_1 = pas d'ordre 1 (en supposant des contraintes linéaires)

p_2 = pas d'ordre 2 (**correction des non linéarités constatées** en $x_1 = x_0 + p_1$)

$$A_1 p_2 = b_1 \text{ avec } \begin{cases} A_1 = \nabla c(x_0)^T \\ b_1 = -c(x_0 + p_1) \end{cases} \rightarrow p_2 = A_1^T (A_1 A_1^T)^{-1} b_1$$

- Pas total : **$p_t = p_1 + p_2$** \rightarrow direction de recherche.



Sommaire

- 1. Bases théoriques**
 - 1.1 Définitions
 - 1.2 Contraintes linéaires
 - 1.3 Contraintes non linéaires
 - 1.4 Conditions d'optimalité**
 - 1.4.1 Dualité
 - 1.4.2 Problème sans contraintes
 - 1.4.3 Problème avec contraintes
 - 1.4.4 Problème linéaire
 - 1.4.5 Problème quadratique
2. Optimisation sans contraintes
3. Optimisation avec contraintes

1.4.1 Dualité

❑ Dualité critère – contraintes

- Critère augmenté
- Lagrangien
- Lagrangien augmenté
- Fonction duale

❑ Problème dual

- Dualité faible
- Saut de dualité
- Point col
- Dualité forte

❑ Programmation linéaire

- Problème primal
- Problème dual

1.4.1 Dualité

Problème avec contraintes égalité

$\min_{x \in \mathbb{R}^n} f(x)$ sous $c(x) = 0 \rightarrow m$ contraintes d'égalité (= contraintes actives)

Dualité

Difficulté de résolution due aux 2 objectifs antagonistes :

- Minimiser le critère $f(x)$
- Satisfaire les contraintes $c(x)=0$
 \rightarrow **Dualité critère-contraintes**

Méthodes duales

Prise en compte des contraintes avec pondération dans la fonction coût

- Critère augmenté \rightarrow pondération = pénalisation des contraintes
- Lagrangien \rightarrow pondération = multiplicateurs de Lagrange
- Lagrangien augmenté \rightarrow pondération = pénalisation + multiplicateurs

\rightarrow Problème sans contraintes plus simple

Réglages des pondérations / **Equivalence au problème avec contraintes**

1.4.1 Critère augmenté

Problème avec contraintes égalité

$\min_{x \in \mathbb{R}^n} f(x)$ sous $c(x) = 0$ → m contraintes d'égalité (= contraintes actives)

Critère augmenté

$$f_{\rho}(x) = f(x) + \frac{1}{2} \rho \|c(x)\|^2$$

ρ = **coefficient de pénalisation** > 0 → Pénalise la violation des contraintes
→ Pondération critère-contraintes

Problème pénalisé sans contraintes

$\min_{x \in \mathbb{R}^n} f_{\rho}(x)$ → Problème **équivalent au problème avec contraintes**
si la pénalisation ρ est assez grande

Problème pénalisé avec contraintes

$\min_{x \in \mathbb{R}^n} f_{\rho}(x)$ sous $c(x) = 0$ → Problème équivalent au problème avec contraintes
Renforce le poids des contraintes dans le critère

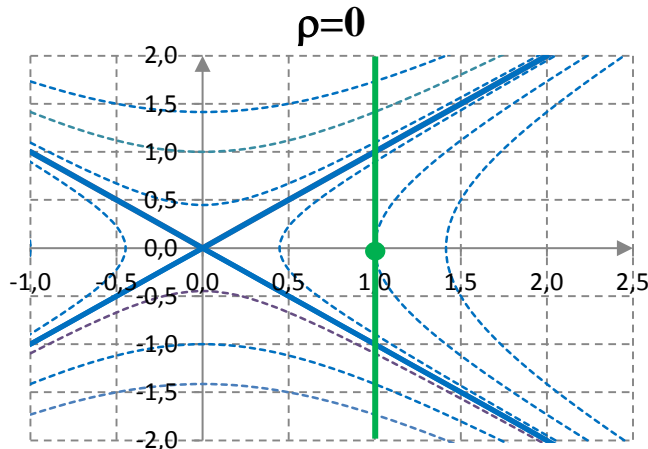
1.4.1 Exemple

Critère augmenté

$$\min_{x_1, x_2} \frac{1}{2}(x_2^2 - x_1^2) \text{ sous } x_1 = 1$$

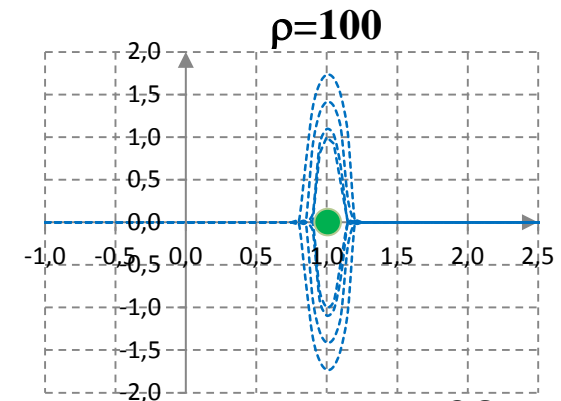
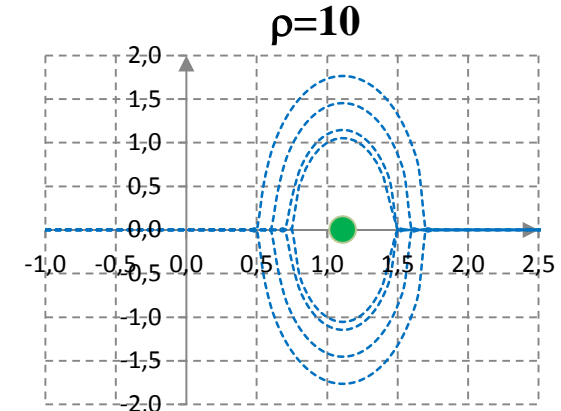
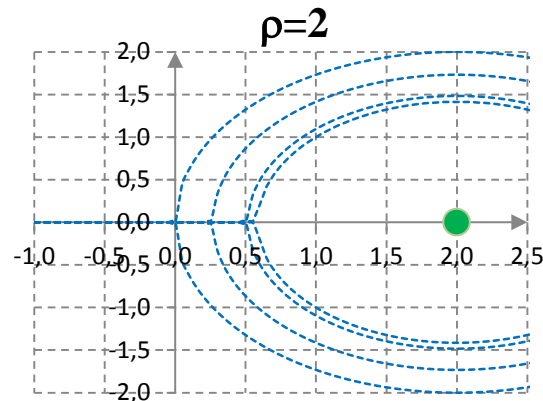
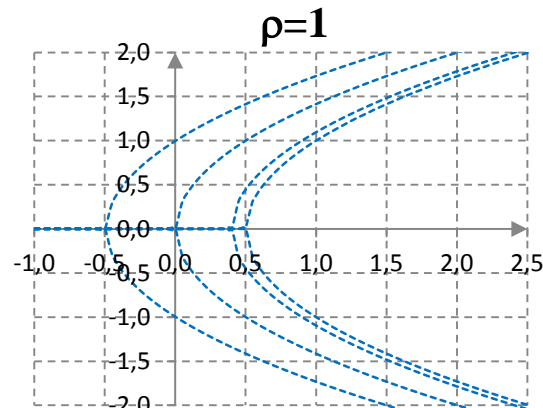
$$f(x) = \frac{1}{2}(x_2^2 - x_1^2) \Rightarrow x^* = (1 \ 0)$$

$$c(x) = x_1 - 1$$



$$f_\rho(x_1, x_2) = \frac{1}{2}(x_2^2 - x_1^2) + \frac{1}{2}\rho(x_1 - 1)^2$$

$$\Rightarrow x^*(\rho) = \left(\frac{\rho}{\rho-1} \quad 0 \right)$$



1.4.1 Lagrangien

Problème avec contraintes égalité et inégalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \quad (\text{PO}) \quad \begin{array}{l} \rightarrow p \text{ contraintes d'égalité} \\ \rightarrow q \text{ contraintes d'inégalité} \end{array}$$

Multiplicateurs de Lagrange

1 multiplicateur par contrainte

- $\lambda \in \mathbb{R}^p \rightarrow$ multiplicateurs des contraintes d'égalité
- $\mu \in \mathbb{R}^q \rightarrow$ multiplicateurs des contraintes d'inégalité

Fonction de Lagrange (ou lagrangien)

Le **lagrangien** du problème (PO) est la fonction L de \mathbb{R}^{n+p+q} dans \mathbb{R}

$$x \in \mathbb{R}^n, \lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q \mapsto L(x, \lambda, \mu) \in \mathbb{R}$$

$$L(x, \lambda, \mu) = f(x) + \lambda^T c_E(x) + \mu^T c_I(x) \quad \Leftrightarrow \quad L(x, \lambda, \mu) = f(x) + \sum_{j=1}^p \lambda_j c_{Ej}(x) + \sum_{j=1}^q \mu_j c_{Ij}(x)$$

- \rightarrow multiplicateurs \approx coefficients de pénalisation des contraintes
- \rightarrow interprétation comme des sensibilités aux niveaux des contraintes

1.4.1 Lagrangien augmenté

Problème pénalisé avec contraintes égalité

- Critère augmenté : coefficient de pénalisation $\rho > 0$

$$\min_{x \in \mathbb{R}^n} f_\rho(x) \text{ sous } c(x) = 0 \quad \text{avec} \quad f_\rho(x) = f(x) + \frac{1}{2}\rho\|c(x)\|^2$$

- Lagrangien du problème pénalisé avec contraintes

$$\begin{aligned} L_\rho(x, \lambda) &= f_\rho(x) + \lambda^T c(x) \\ &= f(x) + \lambda^T c(x) + \frac{1}{2}\rho\|c(x)\|^2 \\ &= L(x, \lambda) + \frac{1}{2}\rho\|c(x)\|^2 \end{aligned}$$

$L_\rho = \text{lagrangien augmenté} = \text{lagrangien initial} + \text{pénalisation des contraintes}$

- Utilisation du lagrangien augmenté
 - Démonstration des conditions suffisantes d'optimalité
 - Algorithme de lagrangien augmenté = suite de minimisations sans contraintes

1.4.1 Exemple

Lagrangien augmenté

- Fonction de 2 variables

$$\min_{x_1, x_2} \frac{1}{2}(x_2^2 - x_1^2) \text{ sous } x_1 = 1$$

→ minimum en $x^* = (1 \ 0)$
 $\lambda^* = 1$

- Critère augmenté**

$$f_\rho(x_1, x_2) = \frac{1}{2}(x_2^2 - x_1^2) + \frac{1}{2}\rho(x_1 - 1)^2$$

→ minimum en $x^*(\rho) = \begin{pmatrix} \frac{\rho}{\rho-1} & 0 \end{pmatrix}$

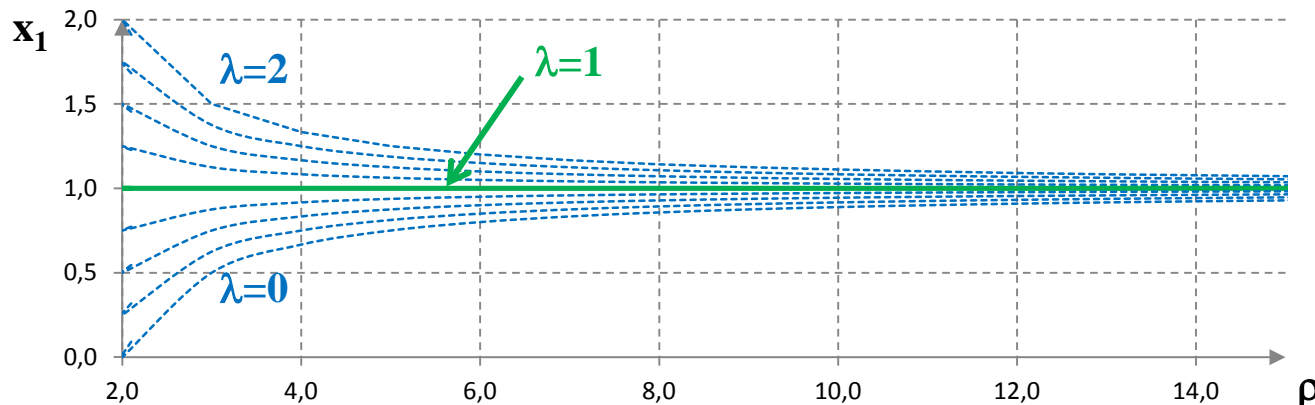
- Lagrangien augmenté**

$$L_\rho(x_1, x_2, \lambda) = \frac{1}{2}(x_2^2 - x_1^2) + \lambda(x_1 - 1) + \frac{1}{2}\rho(x_1 - 1)^2$$

→ minimum en $x^*(\rho, \lambda) = \begin{pmatrix} \frac{\rho - \lambda}{\rho - 1} & 0 \end{pmatrix}$

Pour $\lambda = \lambda^* = 1$, le minimum sans contrainte du lagrangien augmenté est la solution x^* du problème initial.

$$x^*(\rho, \lambda^*) = \begin{pmatrix} \frac{\rho - 1}{\rho - 1} & 0 \end{pmatrix} = (1 \ 0)$$



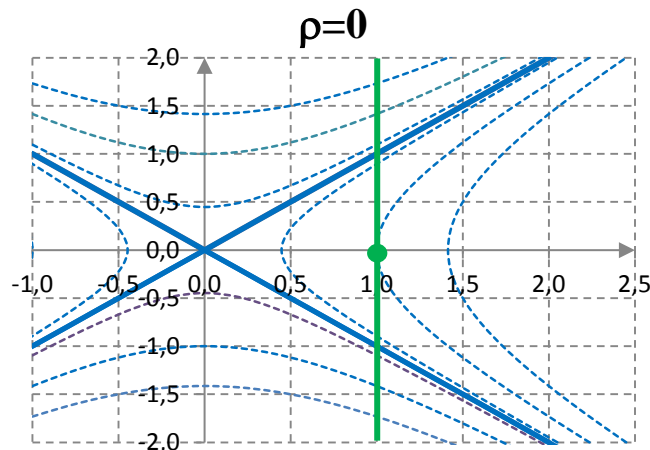
1.4.1 Exemple

Lagrangien augmenté

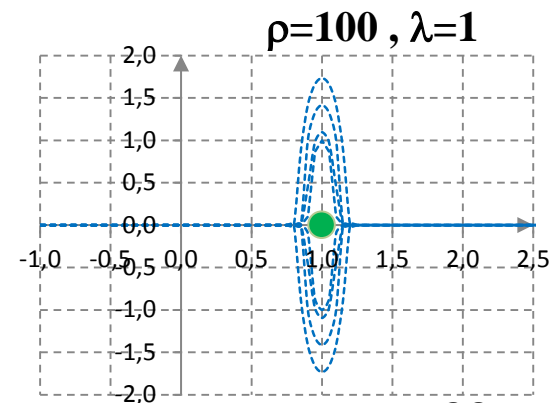
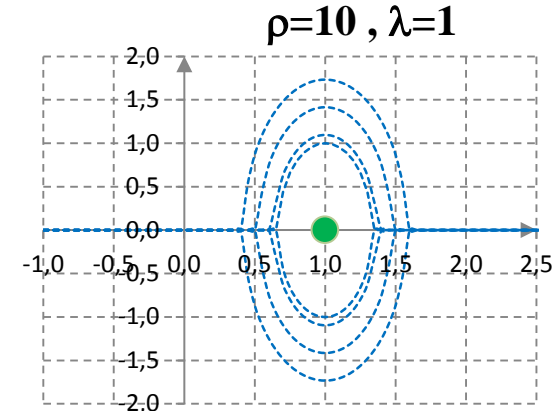
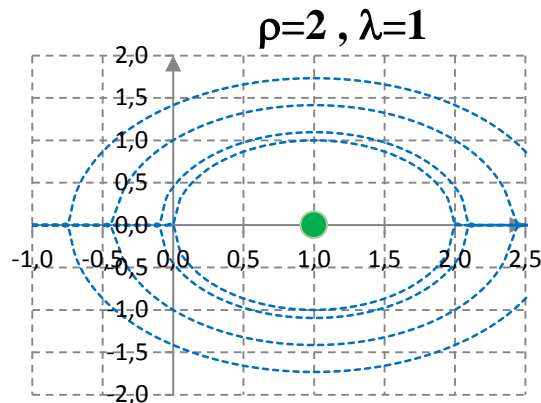
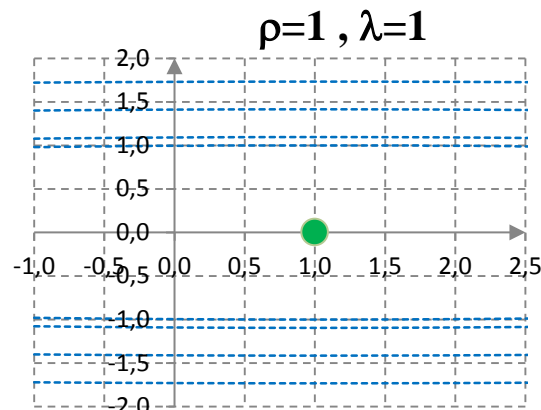
$$\min_{x_1, x_2} \frac{1}{2}(x_2^2 - x_1^2) \text{ sous } x_1 = 1$$

$$f(x) = \frac{1}{2}(x_2^2 - x_1^2) \Rightarrow x^* = (1 \ 0)$$

$$c(x) = x_1 - 1 \quad \lambda^* = 1$$



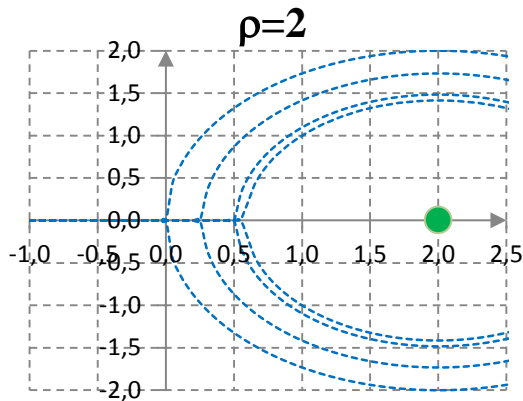
$$L_\rho(x_1, x_2, \lambda) = \frac{1}{2}(x_2^2 - x_1^2) + \lambda(x_1 - 1) + \frac{1}{2}\rho(x_1 - 1)^2 \Rightarrow x^*(\rho) = \begin{pmatrix} \frac{\rho - \lambda}{\rho - 1} & 0 \end{pmatrix}$$



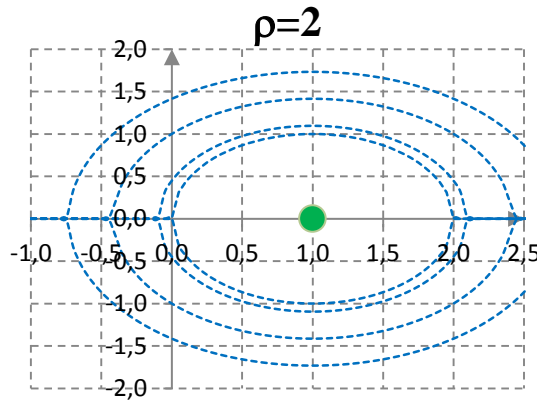
1.4.1 Exemple

Lagrangien augmenté

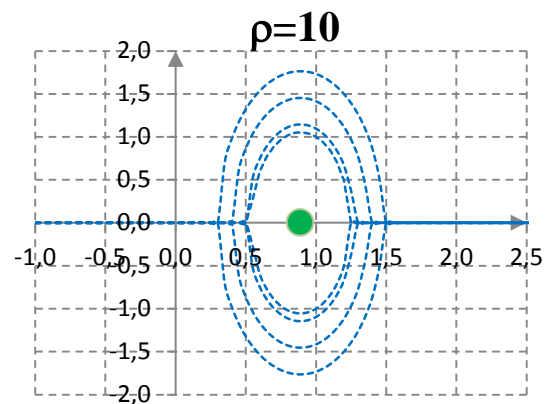
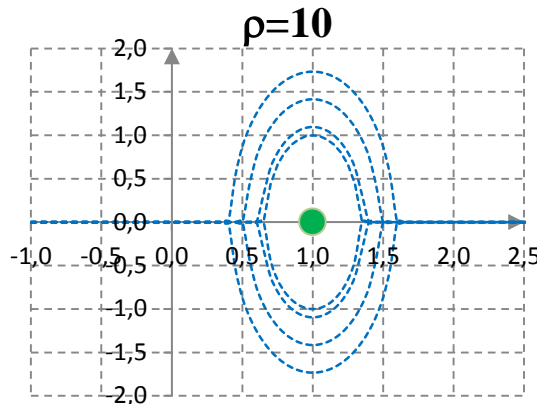
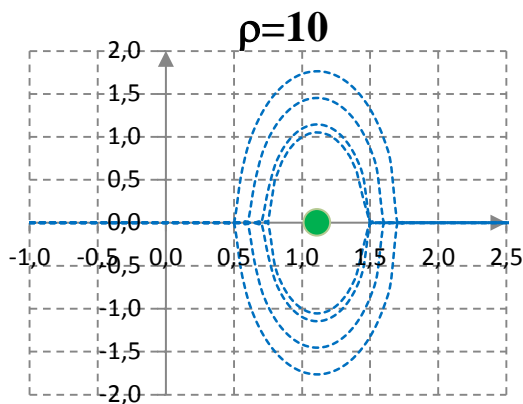
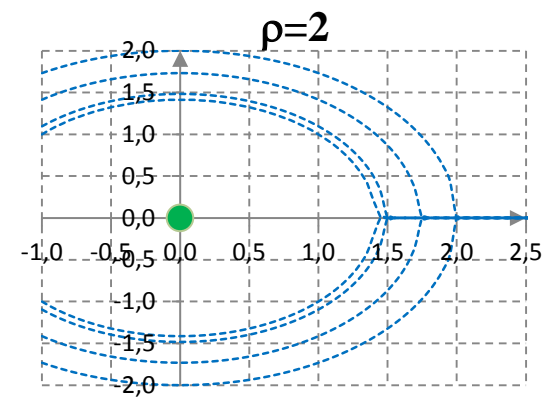
lignes de niveau pour $\lambda = 0$



lignes de niveau pour $\lambda = 1$



lignes de niveau pour $\lambda = 10$



1.4.1 Fonction duale

Fonction duale

La **fonction duale** du problème (PO) est la fonction w de \mathbb{R}^{p+q} dans \mathbb{R}

$$w(\lambda, \mu) = \min_{x \in \mathbb{R}^n} L(x, \lambda, \mu)$$

→ Minimisation du lagrangien à λ et μ fixés

x = **variables primales**

λ et μ = **variables duales**

- Domaine de w : $X_w = \{\lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q / w(\lambda, \mu) > -\infty\}$ → w bornée

Concavité - Convexité

- La fonction duale w est concave
- Le domaine X_w est convexe

Preuve : on note : $\gamma = (\lambda, \mu)$

- $L(x, \alpha\gamma_1 + (1-\alpha)\gamma_2) = \alpha L(x, \gamma_1) + (1-\alpha)L(x, \gamma_2)$ car L linéaire en λ et μ
 $\Rightarrow w(\alpha\gamma_1 + (1-\alpha)\gamma_2) \geq \alpha w(\gamma_1) + (1-\alpha)w(\gamma_2)$ pour le minimum/ x de chaque membre
 → w concave
- Si γ_1 et $\gamma_2 \in X_w$, $w(\alpha\gamma_1 + (1-\alpha)\gamma_2) \geq \alpha w(\gamma_1) + (1-\alpha)w(\gamma_2) > -\infty$
 $\Rightarrow \alpha\gamma_1 + (1-\alpha)\gamma_2 \in X_w$
 → X_w convexe

1.4.1 Problème dual

Problème dual

- Problème primal : $\min_{x \in \mathbb{R}^n} f(x)$ sous $\begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$
 - Fonction duale : $w(\lambda, \mu) = \min_{x \in \mathbb{R}^n} L(x, \lambda, \mu)$ $\rightarrow x(\lambda, \mu) / \nabla_x L(x, \lambda, \mu) = 0$
 - Domaine de w : $X_w = \{\lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q / w(\lambda, \mu) > -\infty\}$ $\rightarrow w$ bornée
 - **Problème dual** : $\max_{\lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q} w(\lambda, \mu)$ sous $(\lambda, \mu) \in X_w, \mu \geq 0$
- $$\Leftrightarrow \max_{x \in \mathbb{R}^n, \lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q} L(x, \lambda, \mu) \text{ sous } \begin{cases} \nabla_x L(x, \lambda, \mu) = 0 \rightarrow x(\lambda, \mu) \\ (\lambda, \mu) \in X_w, \mu \geq 0 \end{cases} \rightarrow \text{dual de Wolfe}$$

Borne sur la fonction duale

- x^* solution du problème primal
- $(\lambda, \mu) \in X_w, \mu \geq 0 \Rightarrow w(\lambda, \mu) \leq f(x^*)$

Preuve :

$$\begin{aligned} w(\lambda, \mu) &= \min_{x \in \mathbb{R}^n} L(x, \lambda, \mu) \leq L(x^*, \lambda, \mu) = f(x^*) + \lambda^T c_E(x^*) + \mu^T c_I(x^*) \\ &= f(x^*) + \mu^T c_I(x^*) \quad \text{car } x^* \text{ admissible} \Rightarrow c_E(x^*) = 0 \\ &\leq f(x^*) \quad \text{car } x^* \text{ admissible} \Rightarrow c_I(x^*) \leq 0 \text{ et } \mu \geq 0 \end{aligned}$$

1.4.1 Exemple

Fonction duale

- **Problème primal** : $\min_{x_1, x_2} \frac{1}{2}(x_2^2 - x_1^2)$ sous $x_1 = 1$

- Lagrangien : $L(x, \lambda) = \frac{1}{2}(x_2^2 - x_1^2) + \lambda(x_1 - 1)$

- Solution : $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \lambda^* = 1$

- **Fonction duale** : $w(\lambda) = \min_x L(x, \lambda) \Rightarrow \frac{\partial L}{\partial x} = 0 \Rightarrow \begin{cases} -x_1 + \lambda = 0 \\ x_2 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = \lambda \\ x_2 = 0 \end{cases}$

$$\Rightarrow w(\lambda) = \frac{1}{2}\lambda^2 - \lambda \quad \text{avec} \quad \begin{cases} x_1 = \lambda \\ x_2 = 0 \end{cases}$$

- **Problème dual** : $\max_{\lambda} w(\lambda) \Rightarrow \frac{\partial w}{\partial \lambda} = 0 \Rightarrow \lambda = 1$

- Solution : $\lambda^* = 1, x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

1.4.1 Dualité faible

Théorème de la dualité faible

- x^* solution du problème primal
 - λ^*, μ^* solution du problème dual
- $$\Rightarrow \boxed{w(\lambda^*, \mu^*) \leq f(x^*)}$$

Preuve : $\forall \lambda, \forall \mu \geq 0, w(\lambda, \mu) \leq f(x^*) \Rightarrow w(\lambda^*, \mu^*) \leq f(x^*)$

Dualité et admissibilité

- Si le problème primal est non borné, le problème dual est non admissible.
- Si le problème dual est non borné, le problème primal est non admissible.

Preuve : en utilisant $w^(\lambda^*, \mu^*) \leq f(x^*)$*

Existence de solutions $x^, \lambda^*, \mu^* \Rightarrow$ fonctions bornées*

Saut de dualité

Le **saut de dualité** est la différence entre la solution du problème primal et du problème dual.

$$\delta = f(x^*) - w(\lambda^*, \mu^*) \geq 0$$

Dans le cas général δ n'est pas nul, il n'est pas équivalent de minimiser f ou maximiser w .

1.4.1 Dualité forte

Point col

$(x^*, \lambda^*, \mu^* \geq 0)$ est un **point col** (ou **point selle**) du lagrangien si

$$\forall (x, \lambda, \mu \geq 0), \begin{cases} L(x^*, \lambda, \mu) \leq L(x^*, \lambda^*, \mu^*) \\ L(x^*, \lambda^*, \mu^*) \leq L(x, \lambda^*, \mu^*) \end{cases} \begin{array}{l} \rightarrow \text{maximisation de } L \text{ par rapport à } (\lambda, \mu) \\ \rightarrow \text{minimisation de } L \text{ par rapport à } (x) \end{array}$$

Caractérisation

$$(x^*, \lambda^*, \mu^* \geq 0) \text{ est un point col du lagrangien si et seulement si } \begin{cases} L(x^*, \lambda^*, \mu^*) = \min_x L(x, \lambda^*, \mu^*) \\ c_E(x^*) = 0 \\ c_I(x^*) \leq 0 \\ \mu^* c_I(x^*) = 0 \end{cases}$$

Théorème de la dualité forte

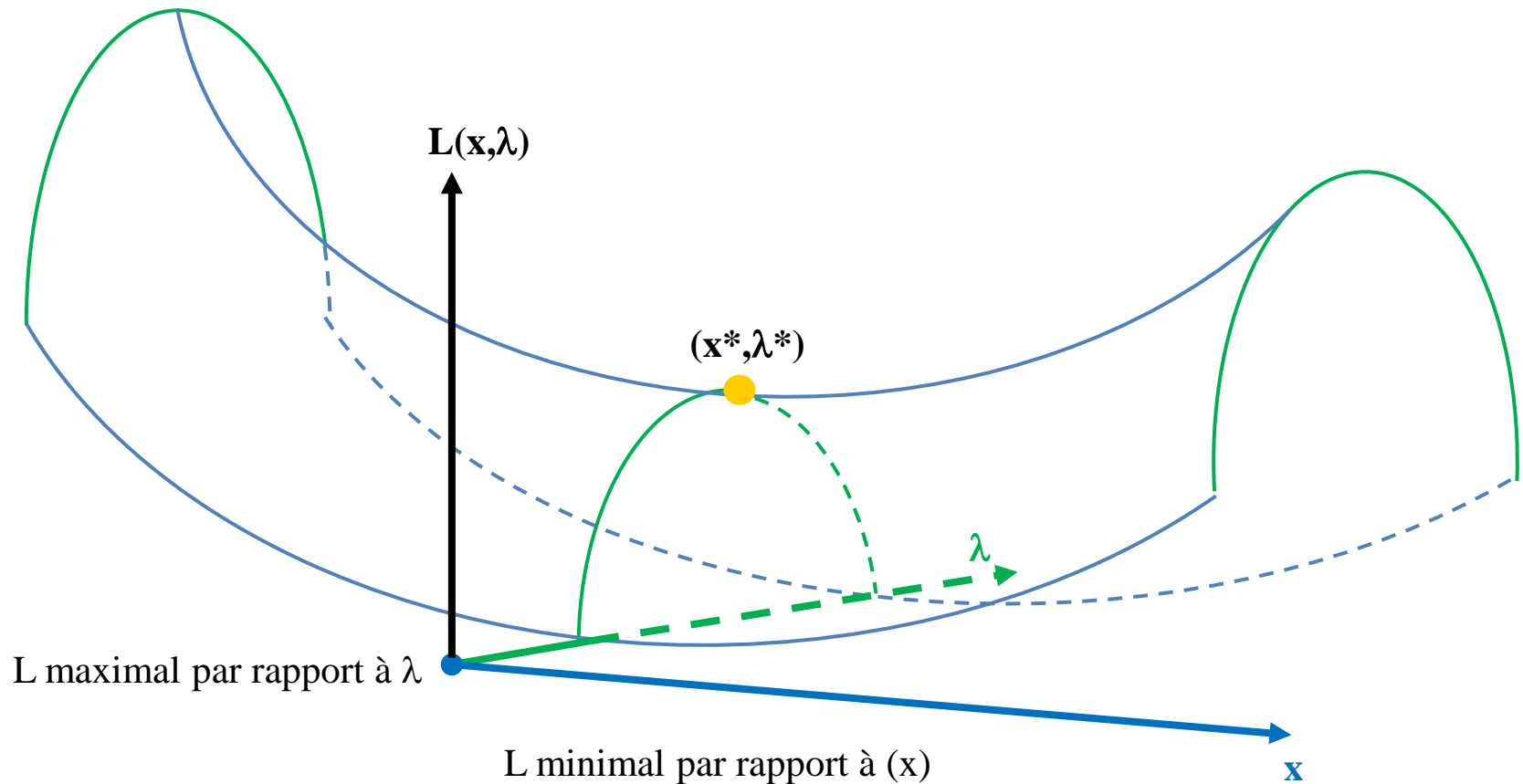
Le lagrangien admet un point col (x^*, λ^*, μ^*) si et seulement si le saut de dualité est nul.

$$(x^*, \lambda^*, \mu^* \geq 0) \text{ un point col} \Leftrightarrow w(\lambda^*, \mu^*) = f(x^*)$$

Il est alors équivalent de minimiser $f(x)$ ou maximiser $w(\lambda, \mu)$.

1.4.1 Dualité forte

Point col (ou point selle)



1.4.1 Dualité forte

Optimum global

Si $(x^*, \lambda^*, \mu^* \geq 0)$ est un point col du lagrangien :

$$\begin{cases} x^* \rightarrow \min_x L(x, \lambda^*, \mu^*) \\ c_E(x^*) = 0, c_I(x^*) \leq 0 \\ \mu^* c_I(x^*) = 0 \end{cases}$$

alors x^* est un **optimum global** du problème primal : $\min_{x \in \mathbb{R}^n} f(x)$ sous $\begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$

En pratique

- Si le lagrangien admet un point col, on peut obtenir l'optimum global x^* .
- Pour un problème non convexe, il n'existe en général pas de point col.

Exemple $\min_x f(x) = -x^2$ sous $\begin{cases} 2x - 1 \leq 0 \\ 0 \leq x \leq 1 \end{cases} \rightarrow \text{solution : } x^* = 0.5$

$$L(x, \mu) = -x^2 + \mu(2x - 1) \text{ avec } \begin{cases} 0 \leq x \leq 1 \\ \mu \geq 0 \end{cases}$$

Point col : $\left\{ \begin{array}{l} \min_x L(x, \mu) \\ 0 \leq x \leq 1 \end{array} \right\} \rightarrow \begin{array}{l} x = 0 \text{ si } \mu > 0.5 \\ x = 1 \text{ si } \mu < 0.5 \end{array}$

$\left\{ \begin{array}{l} 2x - 1 \leq 0 \\ \mu(2x - 1) = 0 \\ \mu \geq 0 \end{array} \right\} \rightarrow \begin{array}{l} \rightarrow x = 0 \\ \rightarrow \mu = 0 < 0.5 \rightarrow x = 1 \end{array} \right\} \rightarrow \text{Il n'existe pas de point col.}$

1.4.1 Programmation linéaire

Problème primal

$$\min_x c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases} \rightarrow \text{problème linéaire sous forme standard}$$

$$\Leftrightarrow \min_x c^T x \text{ sous } \begin{cases} b - Ax = 0 \\ -x \leq 0 \end{cases} \begin{array}{l} \rightarrow \text{multiplicateur } \lambda \\ \rightarrow \text{multiplicateur } \mu \end{array}$$

- Fonction de Lagrange :
$$L(x, \lambda, \mu) = c^T x + \lambda^T (b - Ax) + \mu^T (-x) \\ = (c - A^T \lambda - \mu)^T x + \lambda^T b \rightarrow \text{linéaire en } x$$
- **Fonction duale** :
$$w(\lambda, \mu) = \min_x L(x, \lambda, \mu)$$
- Domaine de définition :
$$X_w = \{(\lambda, \mu) / w(\lambda, \mu) > -\infty\}$$
- La fonction duale n'est définie que si $L(x, \lambda, \mu)$ est borné inférieurement.
 $L(x, \lambda, \mu)$ est linéaire en $x \rightarrow$ Le coefficient de x doit être nul.

$$(\lambda, \mu) \in X_w \Rightarrow c - A^T \lambda - \mu = 0$$

$$\Rightarrow L(x, \lambda, \mu) = \lambda^T b \Rightarrow \boxed{w(\lambda, \mu) = \lambda^T b}$$

1.4.1 Programmation linéaire

Problème dual

$$\max_{\lambda, \mu} w(\lambda, \mu) \text{ sous } \begin{cases} (\lambda, \mu) \in X_w \\ \mu \geq 0 \end{cases}$$

$$\Leftrightarrow \max_{\lambda, \mu} \lambda^T b \text{ sous } \begin{cases} c - A^T \lambda - \mu = 0 \\ \mu \geq 0 \end{cases} \rightarrow \text{ne dépend pas de } \mu$$

$$\Leftrightarrow \max_{\lambda} b^T \lambda \text{ sous } c - A^T \lambda \geq 0 \rightarrow \text{nouveau problème linéaire en } \lambda$$

- Le problème dual est également un problème linéaire dont la variable est λ .
 On met le problème dual sous forme standard en notant la variable y au lieu de λ

$$\min_y -b^T y \text{ sous } A^T y - c \leq 0 \rightarrow \text{multiplicateur } v$$

On peut ensuite définir les fonctions associées à ce problème linéaire.

- Fonction de Lagrange notée $L_d(y, v)$:
$$L_d(y, v) = -b^T y + v^T (A^T y - c) = (Av - b)^T y - v^T c \rightarrow \text{bornée si } Av - b = 0$$
- Fonction duale notée $w_d(v)$:**
$$w_d(v) = \min_y L_d(y, v) = -v^T c \quad \text{si } Av - b = 0$$

1.4.1 Programmation linéaire

Problème dual du problème dual

Le problème dual admet lui-même pour dual :

$$\max_v w_d(v) \text{ sous } \begin{cases} v \in X_{w_d} \\ v \geq 0 \end{cases}$$

$$\Leftrightarrow \max_v -v^T c \text{ sous } \begin{cases} Av = b \\ v \geq 0 \end{cases} \Leftrightarrow \min_x c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases} \rightarrow \text{identique au problème primal}$$

- **Le problème dual du problème dual est le problème primal.**

- Pour un problème linéaire, il est équivalent de résoudre le problème primal ou problème dual. Les solutions du problème primal et du problème dual ont le même coût \rightarrow **dualité forte**

Solutions possibles

		Dual		
		Optimum fini	Optimum infini	Sans solution
Primal	Optimum fini	dualité forte	impossible	impossible
	Optimum infini	impossible	impossible	dualité faible
	Sans solution	impossible	dualité faible	contraintes incompatibles

1.4.1 Programmation linéaire

Correspondances primal – dual

- Problème primal (P) sous forme standard :

$$(P) \quad \min_x c^T x \quad \text{sous} \quad \begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

- Problème dual (D) du problème (P) :

$$(D) \quad \max_y b^T y \quad \text{sous} \quad A^T y \leq c$$

- Le nombre de variables de (P) est égal au nombre de contraintes de (D).
- Le nombre de contraintes de (P) est égal au nombre de variables de (D).
- La matrice des contraintes de (D) est la transposée de la matrice des contraintes de (P).

- Une variable $x_j \geq 0$ de coût c_j donne une contrainte \leq de niveau c_j :

$$\begin{cases} c_j x_j \\ x_j \geq 0 \end{cases} \rightarrow \sum_{i=1}^m a_{ij} y_i \leq c_j$$

- Une contrainte = de niveau b_i donne une variable $y_j \in \mathbb{R}$ de coût b_i :

$$\sum_{j=1}^n a_{ij} x_j = b_i \rightarrow \begin{cases} b_i y_i \\ y_i \in \mathbb{R} \end{cases}$$

→ généralisation à un problème linéaire quelconque (signe des variables, sens des contraintes)

1.4.1 Programmation linéaire

Correspondance primal-dual

- **Problème primal (P)**

$$\min_{\substack{x_1 \in \mathbb{R}^{n_1} \\ x_2 \in \mathbb{R}^{n_2} \\ x_3 \in \mathbb{R}^{n_3}}} c_1^T x_1 + c_2^T x_2 + c_3^T x_3 \quad \text{sous} \quad \begin{cases} A_1 x_1 + B_1 x_2 + C_1 x_3 = b_1, & b_1 \in \mathbb{R}^{m_1} & \rightarrow m_1 \text{ égalités} \\ A_2 x_1 + B_2 x_2 + C_2 x_3 \leq b_2, & b_2 \in \mathbb{R}^{m_2} & \rightarrow m_2 \text{ inégalités inférieur} \\ A_3 x_1 + B_3 x_2 + C_3 x_3 \geq b_3, & b_3 \in \mathbb{R}^{m_3} & \rightarrow m_3 \text{ inégalités supérieur} \\ x_1 \geq 0 & \rightarrow n_1 \text{ variables positives} \\ x_2 \leq 0 & \rightarrow n_2 \text{ variables négatives} \\ x_3 \in \mathbb{R}^{n_3} & \rightarrow n_3 \text{ variables libres} \end{cases}$$

- **Problème dual (D)**

$$\max_{\substack{y_1 \in \mathbb{R}^{m_1} \\ y_2 \in \mathbb{R}^{m_2} \\ y_3 \in \mathbb{R}^{m_3}}} b_1^T y_1 + b_2^T y_2 + b_3^T y_3 \quad \text{sous} \quad \begin{cases} A_1^T y_1 + A_2^T y_2 + A_3^T y_3 \leq c_1, & c_1 \in \mathbb{R}^{n_1} & \rightarrow n_1 \text{ inégalités inférieur} \\ B_1^T y_1 + B_2^T y_2 + B_3^T y_3 \geq c_2, & c_2 \in \mathbb{R}^{n_2} & \rightarrow n_2 \text{ inégalités supérieur} \\ C_1^T y_1 + C_2^T y_2 + C_3^T y_3 = c_3, & c_3 \in \mathbb{R}^{n_3} & \rightarrow n_3 \text{ égalités} \\ y_1 \in \mathbb{R}^{m_1} & \rightarrow m_1 \text{ variables libres} \\ y_2 \leq 0 & \rightarrow m_2 \text{ variables négatives} \\ y_3 \geq 0 & \rightarrow m_3 \text{ variables positives} \end{cases}$$

1.4.1 Programmation linéaire

Correspondance primal-dual

Preuve

- *Lagrangien du problème primal (P)*

$$\begin{aligned}
 L(x_1, x_2, x_3, \lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2) = & c_1^T x_1 + c_2^T x_2 + c_3^T x_3 \\
 & + \lambda_1^T (b_1 - A_1 x_1 - B_1 x_2 - C_1 x_3) & \rightarrow m_1 \text{ multiplicateurs } \lambda_1 \\
 & + \lambda_2^T (A_2 x_1 + B_2 x_2 + C_2 x_3 - b_2) & \rightarrow m_2 \text{ multiplicateurs } \lambda_2 \geq 0 \\
 & + \lambda_3^T (b_3 - A_3 x_1 - B_3 x_2 - C_3 x_3) & \rightarrow m_3 \text{ multiplicateurs } \lambda_3 \geq 0 \\
 & - \mu_1^T x_1 + \mu_2^T x_2 & \rightarrow n_1 \text{ multiplicateurs } \mu_1 \geq 0 \\
 & & \rightarrow n_2 \text{ multiplicateurs } \mu_2 \geq 0
 \end{aligned}$$

- *On regroupe les termes en x_1, x_2, x_3 :*

$$\begin{aligned}
 L(x_1, x_2, x_3, \lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2) = & b_1^T \lambda_1 - b_2^T \lambda_2 + b_3^T \lambda_3 \\
 & + \left(c_1 - A_1^T \lambda_1 + A_2^T \lambda_2 - A_3^T \lambda_3 - \mu_1 \right)^T x_1 & \rightarrow \text{linéaire en } x \\
 & + \left(c_2 - B_1^T \lambda_1 + B_2^T \lambda_2 - B_3^T \lambda_3 + \mu_2 \right)^T x_2 \\
 & + \left(c_3 - C_1^T \lambda_1 + C_2^T \lambda_2 - C_3^T \lambda_3 \right)^T x_3
 \end{aligned}$$

- *La fonction duale est définie par : $w(\lambda, \mu) = \min_x L(x, \lambda, \mu)$ \rightarrow bornée si les coefficients de x_1, x_2, x_3 sont nuls*

1.4.1 Programmation linéaire

Correspondance primal-dual

- $$L(x_1, x_2, x_3, \lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2) = b_1^T \lambda_1 - b_2^T \lambda_2 + b_3^T \lambda_3$$

$$+ \left(c_1 - A_1^T \lambda_1 + A_2^T \lambda_2 - A_3^T \lambda_3 - \mu_1 \right)^T x_1$$

$$+ \left(c_2 - B_1^T \lambda_1 + B_2^T \lambda_2 - B_3^T \lambda_3 + \mu_2 \right)^T x_2$$

$$+ \left(c_3 - C_1^T \lambda_1 + C_2^T \lambda_2 - C_3^T \lambda_3 \right)^T x_3$$
- $$L \text{ bornée} \Rightarrow \begin{cases} c_1 - A_1^T \lambda_1 + A_2^T \lambda_2 - A_3^T \lambda_3 - \mu_1 = 0 \\ c_2 - B_1^T \lambda_1 + B_2^T \lambda_2 - B_3^T \lambda_3 + \mu_2 = 0 \\ c_3 - C_1^T \lambda_1 + C_2^T \lambda_2 - C_3^T \lambda_3 = 0 \end{cases} \text{ avec } \begin{cases} \mu_1, \mu_2 \geq 0 \\ \lambda_2, \lambda_3 \geq 0 \end{cases}$$
- $$\text{En posant : } \begin{cases} y_1 = \lambda_1 \\ y_2 = -\lambda_2 \leq 0 \\ y_3 = \lambda_3 \geq 0 \end{cases} \Rightarrow \begin{cases} A_1^T y_1 + A_2^T y_2 + A_3^T y_3 = c_1 + \mu_1 \leq c_1 & \text{car } \mu_1 \geq 0 \\ B_1^T y_1 + B_2^T y_2 + B_3^T y_3 = c_2 - \mu_2 \geq c_2 & \text{car } \mu_2 \geq 0 \\ C_1^T y_1 + C_2^T y_2 + C_3^T y_3 = c_3 \end{cases}$$
- $$\text{Fonction duale : } w(y_1, y_2, y_3) = L(\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2) \Rightarrow w(y_1, y_2, y_3) = b_1^T y_1 + b_2^T y_2 + b_3^T y_3$$
- $$\text{Problème dual : } \max_{y_1, y_2, y_3} w(y_1, y_2, y_3) = b_1^T y_1 + b_2^T y_2 + b_3^T y_3 \quad \text{sous } \begin{cases} A_1^T y_1 + A_2^T y_2 + A_3^T y_3 \leq c_1 \\ B_1^T y_1 + B_2^T y_2 + B_3^T y_3 \geq c_2 \\ C_1^T y_1 + C_2^T y_2 + C_3^T y_3 = c_3 \end{cases}$$

1.4.1 Exemple

Correspondance primal-dual

- **Problème primal (P)**

$$(P) \quad \min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3 \quad \text{sous} \quad \begin{cases} -x_1 + 3x_2 = 5 \\ 2x_1 - x_2 + 3x_3 \geq 6 \\ x_3 \leq 4 \\ x_1 \geq 0, x_2 \leq 0, x_3 \in \mathbb{R} \end{cases}$$

- **Problème dual (D)**

$$(D) \quad \max_{y_1, y_2, y_3} 5y_1 + 6y_2 + 4y_3 \quad \text{sous} \quad \begin{cases} -y_1 + 2y_2 \leq 1 \\ 3y_1 - y_2 \geq 2 \\ 3y_2 + y_3 = 3 \\ y_1 \in \mathbb{R}, y_2 \geq 0, y_3 \leq 0 \end{cases}$$

$$\Leftrightarrow \min_{y_1, y_2, y_3} -5y_1 - 6y_2 - 4y_3 \quad \text{sous} \quad \begin{cases} y_1 - 2y_2 \geq -1 \\ -3y_1 + y_2 \leq -2 \\ -3y_2 - y_3 = -3 \\ y_1 \in \mathbb{R}, y_2 \geq 0, y_3 \leq 0 \end{cases}$$

- Problème dual du dual : on retrouve le problème primal (P)

1.4.2 Problème sans contraintes

- ☐ Conditions nécessaires d'optimalité locale
- ☐ Conditions suffisantes d'optimalité locale
- ☐ Méthode pratique
- ☐ Exemples

1.4.2 Conditions nécessaires

Problème sans contraintes

$$\min_{x \in \mathbb{R}^n} f(x)$$

Conditions nécessaires

$$x^* \text{ minimum local} \Rightarrow \begin{cases} \nabla f(x^*) = 0 \\ \nabla^2 f(x^*) \geq 0 \end{cases} \begin{array}{l} \rightarrow \text{ordre 1 : point critique ou stationnaire} \\ \rightarrow \text{ordre 2 : hessien semi-défini positif} \end{array}$$

Preuve : avec le théorème de Taylor

- *Ordre 1 :* $f(x^* + d) = f(x^*) + \nabla f(x^*)^T d + o(\|d\|)$

Si $\nabla f(x^) \neq 0$, on peut trouver d petit tel que $\nabla f(x^*)^T d < 0 \Rightarrow f(x^* + d) < f(x^*)$*

- *Ordre 2 :*
$$\begin{aligned} f(x^* + d) &= f(x^*) + \nabla f(x^*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \\ &= f(x^*) + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \quad \text{car } \nabla f(x^*) = 0 \end{aligned}$$

Si $\nabla^2 f(x^)$ non semi-définie positive,*

on peut trouver d petit tel que $d^T \nabla^2 f(x^) d < 0 \Rightarrow f(x^* + d) < f(x^*)$*

1.4.2 Conditions suffisantes

Problème sans contraintes

$$\min_{x \in \mathbb{R}^n} f(x)$$

Conditions suffisantes

$$\begin{cases} \nabla f(x^*) = 0 & \rightarrow \text{ordre 1 : point critique ou stationnaire} \\ \nabla^2 f(x^*) > 0 & \rightarrow \text{ordre 2 : hessien défini positif} \end{cases} \Rightarrow x^* \text{ minimum local}$$

Preuve : par l'absurde

Si x^ n'est pas un minimum local, on peut trouver d petit tel que $f(x^*+d) < f(x^*)$*

Théorème de Taylor à l'ordre 2 :

$$\begin{aligned} f(x^*+d) &= f(x^*) + \nabla f(x^*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \\ &= f(x^*) + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|^2) \quad \text{car } \nabla f(x^*) = 0 \end{aligned}$$

$$f(x^*+d) < f(x^*) \Rightarrow d^T \nabla^2 f(x^*) d < 0 \quad \rightarrow \text{contredit l'hypothèse } \nabla^2 f(x^*) \text{ définie positive}$$

1.4.2 Méthode pratique

Problème sans contraintes

$$\min_{x \in \mathbb{R}^n} f(x)$$

- Condition nécessaire du 1^{er} ordre : $\nabla f(x^*) = 0$ → **point critique ou stationnaire**
- Condition nécessaire du 2^{eme} ordre : $\nabla^2 f(x^*) \geq 0$ → plus difficile à vérifier
suffisante $\nabla^2 f(x^*) > 0$

Méthode pratique

- Recherche des **points stationnaires** en résolvant : $\nabla f(x^*) = 0$
Un point stationnaire peut être un minimum local, un maximum local ou un point selle.
- Vérification de la condition d'ordre 2 : calcul des dérivées secondes
valeurs propres du hessien ≥ 0
→ garantit l'obtention d'un minimum local x^*

Minimum global

x^* minimum local

- **f convexe** $\Rightarrow x^*$ minimum global
- f strictement convexe $\Rightarrow x^*$ unique minimum global
- **f quelconque** (cas général) \Rightarrow **On ne peut pas vérifier que x^* est un minimum global.**

1.4.2 Exemples

Exemple 1

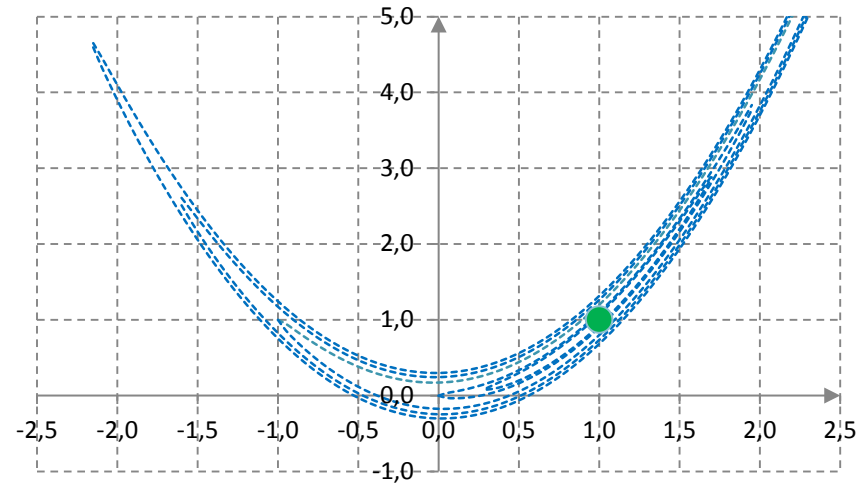
Fonction de Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Gradient :

$$\nabla f(x_1, x_2) = \begin{pmatrix} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 \\ 200x_2 - 200x_1^2 \end{pmatrix}$$
- Hessian :

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}$$
- **Point stationnaire** : $\nabla f(x_1, x_2) = 0 \Rightarrow \begin{cases} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 = 0 \\ 200x_2 - 200x_1^2 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = 1 \\ x_2 = x_1^2 \end{cases} \Rightarrow x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- **Valeurs propres du hessien** : $\nabla^2 f(x^*) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix} \rightarrow \begin{cases} \sigma_1 = 1001.60 \\ \sigma_2 = 0.39936 \end{cases}$
- Condition d'ordre 2 : $\nabla^2 f(x^*)$ est défini positif
- x^* vérifie les conditions suffisantes de minimum local (strict)
 x^* est un **minimum local** de f



1.4.2 Exemples

Exemple 2

Fonction : $f(x_1, x_2) = -x_1^4 - x_2^4$

- Gradient : $\nabla f(x_1, x_2) = \begin{pmatrix} -4x_1^3 \\ -4x_2^3 \end{pmatrix}$
- Hessian : $\nabla^2 f(x_1, x_2) = \begin{pmatrix} -12x_1^2 & 0 \\ 0 & -12x_2^2 \end{pmatrix}$
- **Point stationnaire** : $\nabla f(x_1, x_2) = 0 \Rightarrow \begin{cases} x_1 = 0 \\ x_2 = 0 \end{cases} \Rightarrow x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- **Valeurs propres du hessien** : $\nabla^2 f(x^*) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \rightarrow \begin{cases} \sigma_1 = 0 \\ \sigma_2 = 0 \end{cases}$
 $\nabla^2 f(x^*)$ est semi-défini positif
- x^* vérifie les conditions nécessaires de minimum local
 x^* ne vérifie pas les conditions suffisantes de minimum local

x^* est en fait un **maximum local** de f : $f(x_1, x_2) = -x_1^4 - x_2^4 \leq 0$
 $\Rightarrow \forall (x_1, x_2), f(x_1, x_2) \leq f(0, 0)$

1.4.2 Exemples

Exemple 3

Fonction : $f(x_1, x_2) = x_1^2 - x_2^3$

- Gradient : $\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1 \\ -3x_2^2 \end{pmatrix}$

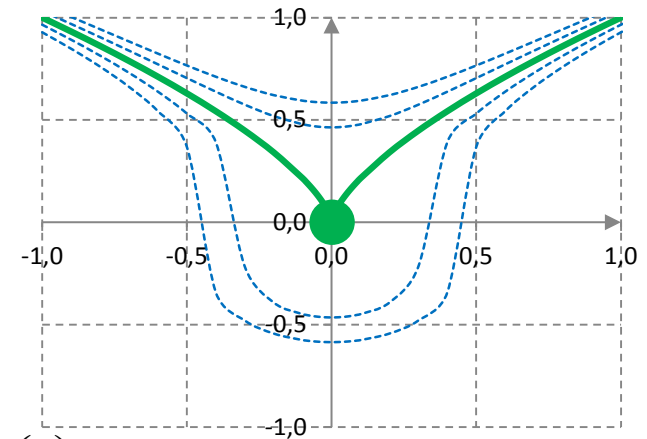
- Hessien : $\nabla^2 f(x_1, x_2) = \begin{pmatrix} 2 & 0 \\ 0 & -6x_2 \end{pmatrix}$

- Point stationnaire** : $\nabla f(x_1, x_2) = 0 \Rightarrow \begin{cases} x_1 = 0 \\ x_2 = 0 \end{cases} \Rightarrow x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

- Valeurs propres du hessien** : $\nabla^2 f(x^*) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \rightarrow \begin{cases} \sigma_1 = 2 \\ \sigma_2 = 0 \end{cases}$
 $\nabla^2 f(x^*)$ est semi-défini positif

- x^* vérifie les conditions nécessaires de minimum local
 x^* ne vérifie pas les conditions suffisantes de minimum local

x^* n'est **ni un minimum ni un maximum local** de f : $f(0, x_2) = -x_2^3 \begin{cases} < 0 \text{ si } x_2 > 0 \\ > 0 \text{ si } x_2 < 0 \end{cases}$



1.4.3 Problème avec contraintes

- ☐ Conditions nécessaires d'optimalité locale
 - Multiplicateurs de Lagrange
 - Conditions KKT
- ☐ Conditions suffisantes d'optimalité locale
- ☐ Interprétation géométrique
- ☐ Méthode pratique
- ☐ Exemples
- ☐ Sensibilité
 - Sensibilité aux niveaux de contrainte
 - Sensibilité aux paramètres de modèle

1.4.3 Conditions nécessaires

Problème avec contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \quad \begin{array}{l} \rightarrow p \text{ contraintes d'égalité} \\ \rightarrow q \text{ contraintes d'inégalité} \end{array}$$

Conditions nécessaires

x^* minimum local $\Rightarrow \nabla f(x^*)^T d \geq 0$ pour toute direction d admissible à la limite en x^*

Méthode directe

Nécessite de connaître l'ensemble des directions admissibles en x^*

- Cas de contraintes linéaires
 - \rightarrow Définition des directions admissibles à partir des directions de base (§1.2.2)
- Cas de contraintes non linéaires
 - \rightarrow Définition des directions admissibles à la limite
 - \rightarrow Pas de caractérisation des directions admissibles dans le cas général
sauf hypothèse de qualification des contraintes : cône des directions (§1.3.1)

Méthode indirecte

A partir des multiplicateurs de Lagrange

\rightarrow **Conditions d'optimalité dans le cas général**

1.4.3 Conditions nécessaires

Problème avec contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \quad \begin{array}{l} \rightarrow p \text{ contraintes d'égalité} \\ \rightarrow q \text{ contraintes d'inégalité} \end{array}$$

Conditions nécessaires

Hypothèse : Contraintes linéairement indépendantes en x^*

x^* minimum local \Rightarrow Il existe un unique $\lambda^* \in \mathbb{R}^p$ et un unique $\mu^* \in \mathbb{R}^q$ tels que :

- **Ordre 1** :
$$\begin{cases} \nabla_x L(x^*, \lambda^*, \mu^*) = 0 \\ \nabla_\lambda L(x^*, \lambda^*, \mu^*) = 0 \\ \nabla_\mu L(x^*, \lambda^*, \mu^*) \leq 0 \\ \mu^* \geq 0 \\ \mu^* c_I(x^*) = 0 \end{cases} \quad \begin{array}{l} \rightarrow \text{conditions nécessaires d'ordre 1} \\ \rightarrow \text{contraintes égalité } c_E(x^*) = 0 \\ \rightarrow \text{contraintes inégalité } c_I(x^*) \leq 0 \\ \\ \rightarrow \text{conditions complémentaires} \end{array}$$
- **Ordre 2** : Pour toute direction d tangente aux contraintes actives ($c(x^*)=0$) :

$$d^T \nabla_{xx}^2 L(x^*, \lambda^*, \mu^*) d \geq 0 \quad \rightarrow \text{conditions nécessaires d'ordre 2}$$

$$\forall d / d^T \nabla c(x^*) = 0$$

\rightarrow **Conditions nécessaires de Karush-Kuhn-Tucker (conditions KKT)**

(1939) (1951)

1.4.3 Conditions nécessaires

Eléments de la démonstration

1) Cas de contraintes linéaires : $c_E(x) = Ax - b = 0$
 $\rightarrow \nabla c_E(x) = A^T$

- On choisit une base $B \in \mathbb{R}^{m \times n}$: $AE = \begin{pmatrix} m & n-m \\ B & N \end{pmatrix} \quad E^T x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$

$$Ax = b \Leftrightarrow Bx_B + Nx_N = b \Leftrightarrow x_B = B^{-1}(b - Nx_N)$$

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } Ax = b \Leftrightarrow \min_{\substack{x_B \in \mathbb{R}^m \\ x_N \in \mathbb{R}^{n-m}}} f(x_B, x_N) \text{ sous } x_B = B^{-1}(b - Nx_N)$$

- On se ramène à un problème sans contrainte (= **problème réduit**)

$$\min_{x_N \in \mathbb{R}^{n-m}} g(x_N) = f(B^{-1}(b - Nx_N), x_N)$$

- Conditions nécessaires d'optimalité du problème réduit

$$x_N^* \text{ minimum local de } g(x_N) \Rightarrow \begin{cases} \nabla g(x_N^*) = 0 \\ \nabla^2 g(x_N^*) \geq 0 \end{cases}$$

\rightarrow Méthode de réduction : fonction réduite $g(x_N)$
gradient réduit $\nabla g(x_N)$
hessien réduit $\nabla^2 g(x_N)$

1.4.3 Conditions nécessaires

Eléments de la démonstration

1) Cas de contraintes linéaires

- *Problème réduit :* $\min_{x_N \in \mathbb{R}^{n-m}} g(x_N) = f(B^{-1}(b - Nx_N), x_N)$
- *Condition nécessaire d'ordre 1 du problème réduit :* $\nabla g(x_N^*) = 0$

$$g(x_N) = f(B^{-1}(b - Nx_N), x_N) \quad \text{avec} \quad \nabla f(x) = \begin{pmatrix} \nabla_B f(x) \\ \nabla_N f(x) \end{pmatrix}$$

$$\Rightarrow \nabla g(x_N) = -(B^{-1}N)^T \nabla_B f(x) + \nabla_N f(x)$$

- *On définit :* $\lambda = -B^{-T} \nabla_B f(x) \Leftrightarrow \nabla_B f(x) + B^T \lambda = 0$
 $\Rightarrow \nabla g(x_N) = N^T \lambda + \nabla_N f(x)$

$$\left. \begin{array}{l} \nabla g(x_N^*) = 0 \Rightarrow \nabla_N f(x^*) + N^T \lambda^* = 0 \\ \text{avec } \nabla_B f(x^*) + B^T \lambda^* = 0 \end{array} \right\} \Rightarrow \nabla f(x^*) + A^T \lambda^* = 0$$

- *On obtient la condition nécessaire d'ordre 1 sur le lagrangien*

$$\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) + \nabla c_E(x^*) \lambda^* = 0 \quad \text{car } \nabla c_E(x^*)^T = A$$

1.4.3 Conditions nécessaires

Eléments de la démonstration

1) Cas de contraintes linéaires

- Condition nécessaire d'ordre 2 du problème réduit : $\nabla^2 g(x_N^*) \geq 0$

$$g(x_N) = f(B^{-1}(b - Nx_N), x_N) \quad \text{avec} \quad \nabla^2 f(x) = \begin{pmatrix} \nabla_{BB}^2 f(x) & \nabla_{BN}^2 f(x) \\ \nabla_{NB}^2 f(x) & \nabla_{NN}^2 f(x) \end{pmatrix}$$

$$\Rightarrow \nabla^2 g(x_N) = (B^{-1}N)^T \nabla_{BB}^2 f(x) (B^{-1}N) - (B^{-1}N)^T \nabla_{BN}^2 f(x) - \nabla_{NB}^2 f(x) (B^{-1}N) + \nabla_{NN}^2 f(x)$$

- Pour $d \in R$ vérifiant $Ad = 0 \Rightarrow d = \begin{pmatrix} d_B \\ d_N \end{pmatrix} = \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix} d_N$, $d_N \in R^{n-m}$

$$d_N^T \nabla^2 g(x_N) d_N = d_N^T (B^{-1}N)^T \nabla_{BB}^2 f(x) (B^{-1}N) d_N - d_N^T (B^{-1}N)^T \nabla_{BN}^2 f(x) d_N \\ - d_N^T \nabla_{NB}^2 f(x) (B^{-1}N) d_N + d_N^T \nabla_{NN}^2 f(x) d_N$$

$$\Rightarrow d_N^T \nabla^2 g(x_N) d_N = d_B^T \nabla_{BB}^2 f(x) d_B - d_B^T \nabla_{BN}^2 f(x) d_N - d_N^T \nabla_{NB}^2 f(x) d_B + d_N^T \nabla_{NN}^2 f(x) d_N$$

$$\Rightarrow d_N^T \nabla^2 g(x_N) d_N = d^T \nabla^2 f(x) d = d^T \nabla_{xx}^2 L(x, \lambda) d \quad \text{car} \quad \nabla_x L(x^*, \lambda^*) = \nabla f(x^*) + A^T \lambda^* \\ \Rightarrow \nabla_{xx}^2 L(x^*, \lambda^*) = \nabla^2 f(x^*)$$

- On obtient la condition nécessaire d'ordre 2 sur le lagrangien

$$\forall d \in R / Ad = 0, \quad d^T \nabla_{xx}^2 L(x^*, \lambda^*) d = d_N^T \nabla^2 g(x_N^*) d_N \geq 0$$

1.4.3 Conditions nécessaires

Eléments de la démonstration

2) *Cas de contraintes non linéaires* : $c_E(x)=0$
 $c_I(x) \leq 0$

On suppose que x^* est un minimum local de $\min_{x \in R^n} f(x)$ sous $\begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$

- On définit pour les contraintes inégalité : $c_I^+(x) = \max(0, c_I(x))$
- On considère une suite de problèmes pénalisés sans contrainte

$$\min_{x \in R^n} f_k(x) = f(x) + \frac{1}{2}k \|c_E(x)\|^2 + \frac{1}{2}k \|c_I^+(x)\|^2 + \frac{1}{2}\alpha \|x^* - x\|^2 \rightarrow \text{minimum local } x_k$$

$k \in N \rightarrow$ pénalisation de la violation des contraintes

$\alpha > 0 \rightarrow$ pénalisation de la distance au minimum

- x_k minimum local de $f_k \Rightarrow f_k(x_k) \leq f_k(x^*) = f(x^*)$

\Rightarrow La suite $f_k(x_k)$ est bornée supérieurement par $f(x^*)$

1.4.3 Conditions nécessaires

Eléments de la démonstration

2) *Cas de contraintes non linéaires* : $c_E(x)=0$
 $c_I(x) \leq 0$

- La suite $f_k(x_k)$ est bornée supérieurement par $f(x^*)$

$$f_k(x_k) = f(x_k) + \frac{1}{2}k\|c_E(x_k)\|^2 + \frac{1}{2}k\|c_I^+(x_k)\|^2 + \frac{1}{2}\alpha\|x^* - x_k\|^2 \leq f(x^*)$$

$$\Rightarrow \begin{cases} k\|c_E(x_k)\|^2 & \text{borné} \\ k\|c_I^+(x_k)\|^2 & \text{borné} \\ \alpha\|x^* - x_k\|^2 & \text{borné} \end{cases} \Rightarrow \begin{cases} \lim_{k \rightarrow \infty} kc_E(x_k) \stackrel{\text{def}}{=} \lambda^* & \rightarrow \text{définit les multiplicateurs } \lambda \\ \lim_{k \rightarrow \infty} kc_I^+(x_k) \stackrel{\text{def}}{=} \mu^* & \rightarrow \text{définit les multiplicateurs } \mu \\ \lim_{k \rightarrow \infty} x_k = x^* & \rightarrow \text{la suite } (x_k) \text{ converge vers } x^* \end{cases}$$

- Multiplicateurs des contraintes inégalité : $c_I^+(x) = \max(0, c_I(x)) \geq 0 \Rightarrow \mu^* \geq 0$
 Contrainte inégalité inactive : $c_I(x) = 0 \Rightarrow c_I(x_k) = 0, k > k_0 \Rightarrow \mu^* = 0$

- On regroupe les contraintes en un seul vecteur pour simplifier l'écriture

$$c(x) = \begin{pmatrix} c_E(x) \\ c_I^+(x) \end{pmatrix} \rightarrow f_k(x) = f(x) + \frac{1}{2}k\|c(x)\|^2 + \frac{1}{2}\alpha\|x^* - x\|^2$$

$$\nabla f_k(x) = \nabla f(x) + k\nabla c(x)c(x) + \alpha(x - x^*)$$

1.4.3 Conditions nécessaires

Eléments de la démonstration

2) *Cas de contraintes non linéaires* : $c_E(x)=0$
 $c_I(x) \leq 0$

- *Condition nécessaire d'optimalité pour le problème sans contrainte*

$$x_k \text{ minimum local de } f_k(x) \Rightarrow \nabla f_k(x_k) = 0 \Rightarrow \nabla f(x_k) + k \nabla c(x_k) c(x_k) + \alpha(x_k - x^*) = 0$$

- *En prémultipliant par $\nabla c(x_k)^T$*

$$\Rightarrow \nabla c(x_k)^T \nabla f(x_k) + k \nabla c(x_k)^T \nabla c(x_k) c(x_k) + \alpha \nabla c(x_k)^T (x_k - x^*) = 0$$
- *Les contraintes sont supposées linéairement indépendantes $\rightarrow \nabla c(x_k)^T \nabla c(x_k)$ inversible*
- *En prémultipliant par $(\nabla c(x_k)^T \nabla c(x_k))^{-1}$*

$$\Rightarrow k c(x_k) = -(\nabla c(x_k)^T \nabla c(x_k))^{-1} \nabla c(x_k)^T (\nabla f(x_k) + \alpha(x_k - x^*))$$
- *En distinguant les contraintes égalité et inégalité*

$$\Rightarrow \begin{cases} k c_E(x_k) = -(\nabla c_E(x_k)^T \nabla c_E(x_k))^{-1} \nabla c_E(x_k)^T (\nabla f(x_k) + \alpha(x_k - x^*)) \\ k c_I^+(x_k) = -(\nabla c_I^+(x_k)^T \nabla c_I^+(x_k))^{-1} \nabla c_I^+(x_k)^T (\nabla f(x_k) + \alpha(x_k - x^*)) \end{cases}$$

1.4.3 Conditions nécessaires

Eléments de la démonstration

2) *Cas de contraintes non linéaires* : $c_E(x)=0$
 $c_I(x) \leq 0$

- On obtient :
$$\begin{cases} kc_E(x_k) = -\left(\nabla c_E(x_k)^T \nabla c_E(x_k)\right)^{-1} \nabla c_E(x_k)^T (\nabla f(x_k) + \alpha(x_k - x^*)) \\ kc_I^+(x_k) = -\left(\nabla c_I^+(x_k)^T \nabla c_I^+(x_k)\right)^{-1} \nabla c_I^+(x_k)^T (\nabla f(x_k) + \alpha(x_k - x^*)) \end{cases}$$
- Lorsque k tend vers l'infini

$$\begin{cases} \lim_{k \rightarrow \infty} kc_E(x_k) \stackrel{\text{def}}{=} \lambda^* = -\left(\nabla c_E(x^*)^T \nabla c_E(x^*)\right)^{-1} \nabla c_E(x^*)^T \nabla f(x^*) \\ \lim_{k \rightarrow \infty} kc_I^+(x_k) \stackrel{\text{def}}{=} \mu^* = -\left(\nabla c_I^+(x^*)^T \nabla c_I^+(x^*)\right)^{-1} \nabla c_I^+(x^*)^T \nabla f(x^*) \end{cases}$$

- En reportant dans l'expression du gradient de f_k en x_k

$$\nabla f_k(x_k) = \nabla f(x_k) + k \nabla c_E(x_k) c_E(x_k) + k \nabla c_I^+(x_k) c_I^+(x_k) + \alpha(x_k - x^*) = 0$$

$$\xrightarrow{k \rightarrow \infty} \nabla f(x^*) + \nabla c_E(x^*) \lambda^* + \nabla c_I^+(x^*) \mu^* = 0$$

$$\Rightarrow \nabla f(x^*) + \nabla c_E(x^*) \lambda^* + \nabla c_I(x^*) \mu^* = 0 \quad \text{car } \mu^* = 0 \text{ pour les inégalités inactives}$$

- On obtient la condition nécessaire d'ordre 1 : $\nabla_x L(x^*, \lambda^*, \mu^*) = 0$

1.4.3 Exemple

Conditions nécessaires

$$\min_{x_1, x_2} \frac{1}{2}(x_2^2 - x_1^2) \text{ sous } x_1 \leq 1$$

$$\text{Lagrangien : } L(x_1, x_2, \mu) = \frac{1}{2}(x_2^2 - x_1^2) + \mu(x_1 - 1)$$

- Conditions nécessaires d'ordre 1**

$$\begin{cases} -x_1 + \mu = 0 \\ x_2 = 0 \\ x_1 \leq 1 \\ \mu \geq 0 \\ \mu(x_1 - 1) = 0 \end{cases} \rightarrow \text{vérifiées en } x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mu^* = 1$$

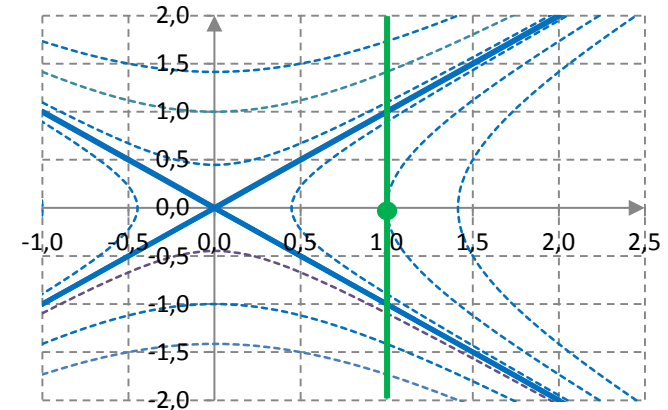
- Conditions nécessaires d'ordre 2**

d direction tangente aux contraintes actives : $d^T \nabla c(x^*) = 0 \Rightarrow \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 0 \Rightarrow d_1 = 0$

$$d^T \nabla_{xx}^2 L(x^*, \mu^*) d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}^T \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = -d_1^2 + d_2^2 = d_2^2 \geq 0$$

$x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mu^* = 1$

 vérifie les conditions nécessaires d'ordre 1 et 2.



1.4.3 Conditions suffisantes

Problème avec contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \quad \begin{array}{l} \rightarrow p \text{ contraintes d'égalité} \\ \rightarrow q \text{ contraintes d'inégalité} \end{array}$$

Conditions suffisantes

S'il existe $x^* \in \mathbb{R}^n$, $\lambda^* \in \mathbb{R}^p$, $\mu^* \in \mathbb{R}^q$ tels que :

- **Ordre 1** :
$$\begin{cases} \nabla_x L(x^*, \lambda^*, \mu^*) = 0 & \rightarrow \text{conditions d'ordre 1} \\ \nabla_\lambda L(x^*, \lambda^*, \mu^*) = 0 & \rightarrow \text{contraintes égalité } c_E(x^*) = 0 \\ \nabla_\mu L(x^*, \lambda^*, \mu^*) \leq 0 & \rightarrow \text{contraintes inégalité } c_I(x^*) \leq 0 \\ \mu^* \geq 0 \\ \mu^* c_I(x^*) = 0 & \rightarrow \text{conditions complémentaires} \\ \mu_k^* > 0 \text{ si } c_{I_k}(x^*) = 0 & \rightarrow \text{contraintes actives : multiplicateur} > 0 \end{cases}$$

- **Ordre 2** : Pour toute direction d tangente aux contraintes actives ($c(x^*)=0$) :

$$d^T \nabla_{xx}^2 L(x^*, \lambda^*, \mu^*) d > 0 \quad \rightarrow \text{conditions d'ordre 2}$$

$$\forall d / d^T \nabla c(x^*) = 0$$

$\Rightarrow x^*$ est un minimum local strict

Remarque : Pas d'hypothèse de qualification des contraintes dans les conditions suffisantes

1.4.3 Conditions suffisantes

Eléments de la démonstration

Cas de contraintes égalité : $c(x)=0$

On suppose que (x^, λ^*) vérifie les conditions suffisantes.*

- *On considère le problème sans contrainte*

$$\min_{x \in \mathbb{R}^n} L_\rho(x, \lambda^*) = L(x, \lambda^*) + \frac{1}{2} \rho \|c(x)\|^2 = f(x) + \lambda^{*T} c(x) + \frac{1}{2} \rho \|c(x)\|^2$$

$L_\rho(x, \lambda) =$ **lagrangien augmenté**

$\rho > 0$ = pénalisation de la violation des contraintes

- $\nabla_x L_\rho(x^*, \lambda^*) = \nabla f(x^*) + \nabla c(x^*) \lambda^* + \rho \nabla c(x^*) c(x^*)$
 $= \nabla f(x^*) + \nabla c(x^*) \lambda^*$ \rightarrow car x^* admissible
 $= \nabla_x L(x^*, \lambda^*) = 0$ \rightarrow par hypothèse sur x^*, λ^*
- $\nabla_{xx}^2 L_\rho(x^*, \lambda^*) = \nabla_{xx}^2 L(x^*, \lambda^*) + \rho \nabla c(x^*) \nabla c(x^*)^T$ \rightarrow définie positive pour ρ assez grand

$\Rightarrow x^*$ est un minimum local du lagrangien augmenté $L_\rho(x, \lambda^*)$ pour $\lambda = \lambda^*$.

- Au voisinage de x^* : $L_\rho(x^*, \lambda^*) \leq L_\rho(x, \lambda^*) \Rightarrow f(x^*, \lambda^*) \leq f(x, \lambda^*) , \forall x / c(x) = 0$
 $\Rightarrow x^*$ est un minimum local de f

1.4.3 Exemple

Conditions suffisantes

$$\min_{x_1, x_2} \frac{1}{2} (x_2^2 - x_1^2) \text{ sous } x_1 \leq 1$$

$$x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mu^* = 1 \text{ vérifie les conditions nécessaires}$$

- Conditions suffisantes d'ordre 1**

Contrainte active \rightarrow multiplicateur > 0

$$x^* - 1 = 0 \quad \mu^* = 1 > 0$$

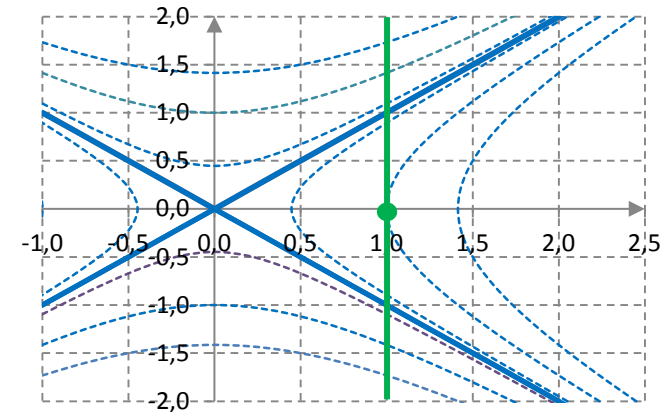
- Conditions suffisantes d'ordre 2**

$$d \text{ direction tangente aux contraintes actives : } d^T \nabla c(x^*) = 0 \Rightarrow \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 0 \Rightarrow d_1 = 0$$

$$d^T \nabla_{xx}^2 L(x^*, \mu^*) d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}^T \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = -d_1^2 + d_2^2 = d_2^2 > 0 \text{ car } d = \begin{pmatrix} 0 \\ d_2 \end{pmatrix} \neq 0 \Rightarrow d_2 \neq 0$$

$x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mu^* = 1$

 vérifie les conditions suffisantes d'ordre 1 et 2 \rightarrow minimum local strict.



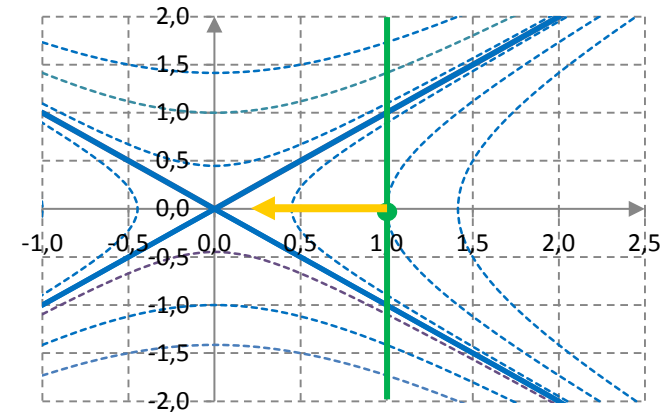
1.4.3 Exemple

Remarque sur la condition d'ordre 2

$$\min_{x_1, x_2} \frac{1}{2} (x_2^2 - x_1^2) \text{ sous } x_1 \leq 1 \rightarrow x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mu^* = 1$$

$$d = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \text{ est une direction admissible en } x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

mais la condition d'ordre 2 ne porte que sur les directions tangentes aux contraintes actives.



Importance de la condition de complémentarité

$$\min_{x_1, x_2} \frac{1}{2} (x_2^2 - x_1^2) \text{ sous } x_1 \leq 0$$

$$x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mu^* = 0 \quad \begin{array}{l} \text{vérifie les conditions suffisantes d'ordre 1 et 2} \\ \text{sauf la condition de complémentarité} \end{array}$$

Si la contrainte active est active, le multiplicateur doit être strictement positif.

$x^* - 1 = 0$ active et $\mu^* = 0$ n'est pas strictement positif

$\rightarrow x^*$ n'est pas un minimum local (f décroît suivant $x_1 < 0$)

1.4.3 Interprétation géométrique

Interprétation

- Condition complémentaire

$$\mu_j c_{I_j}(x) = 0, \quad j = 1, \dots, q \Rightarrow \begin{cases} \mu_j = 0 & \rightarrow \text{sensibilité nulle} \\ \text{ou} \\ c_{I_j}(x) = 0 & \rightarrow \text{contrainte active} \end{cases}$$

- Condition d'ordre 1

$$\begin{aligned} \nabla_x L(x, \lambda, \mu) = 0 &\Rightarrow \nabla f(x) + \nabla c_E(x) \cdot \lambda + \nabla c_I(x) \cdot \mu = 0 \\ &\Rightarrow -\nabla f(x) = \nabla c_E(x) \cdot \lambda + \nabla c_I(x) \cdot \mu \\ &\Rightarrow -\nabla f(x) = \nabla c(x) \cdot v \quad \rightarrow \text{contraintes actives } c(x) \end{aligned}$$

La direction $-\nabla f(x)$ est la direction de plus forte descente en x .

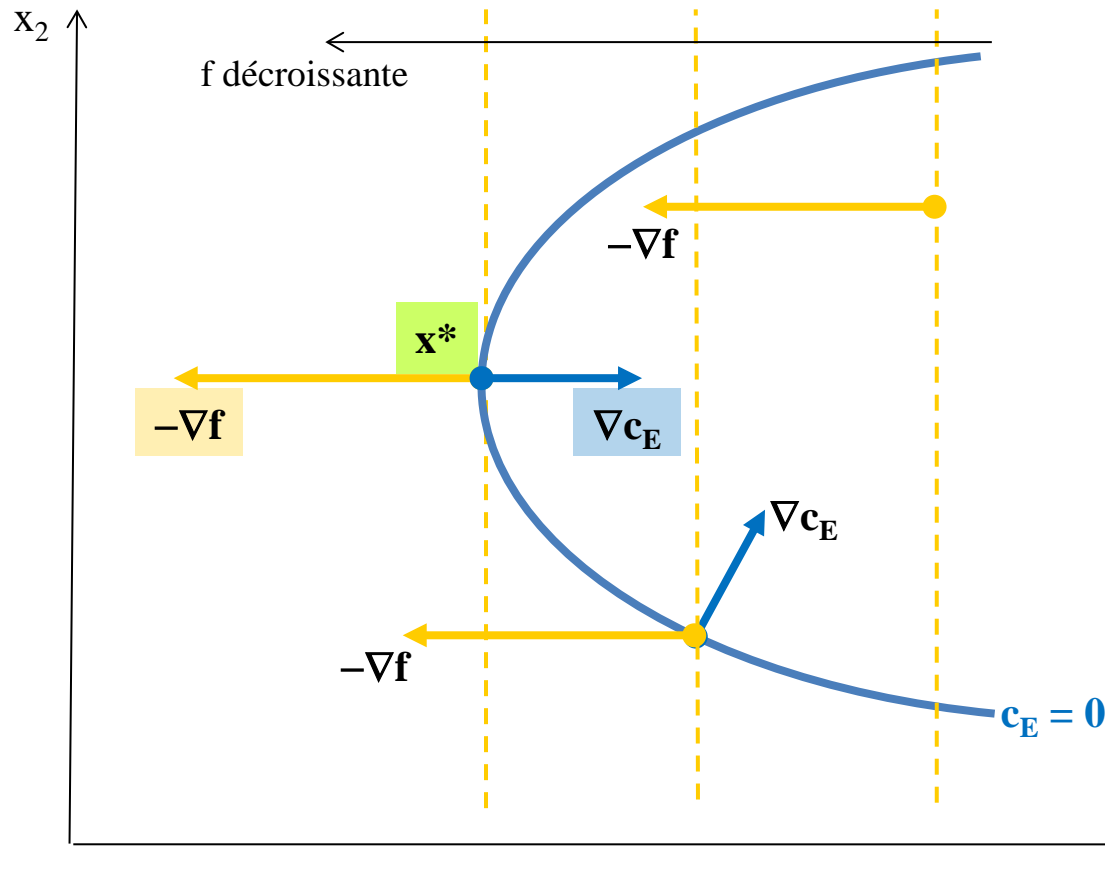
Les directions $\nabla c(x)$ sont orthogonales à l'hyperplan tangent aux contraintes actives en x .

Equation de l'hyperplan tangent aux contraintes actives en x : $d^T \nabla c(x) = 0$

- Les déplacements admissibles (dans l'hyperplan tangent) sont orthogonaux au gradient.
- **Déplacements suivant les lignes de niveau de f , sans diminution du critère.**

1.4.3 Interprétation géométrique

Fonction de 2 variables – 1 contrainte égalité



$$\min_{x_1, x_2} f(x_1, x_2) \text{ sous } c_E(x_1, x_2) = 0$$

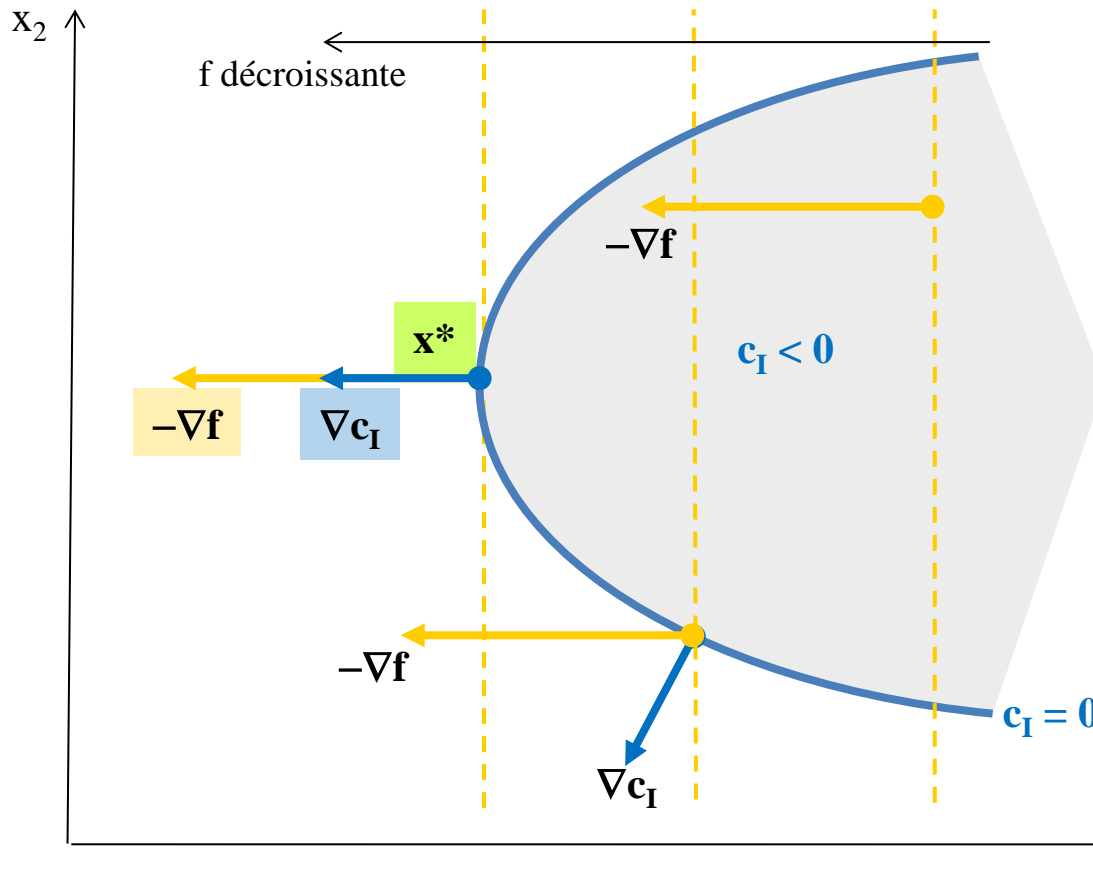
$$\nabla f(x^*) + \lambda \nabla c_E(x^*) = 0$$

Sur le schéma :

- $f(x_1, x_2) = x_1$ à minimiser
- $\lambda < 0$ ($\lambda \approx -2$)

1.4.3 Interprétation géométrique

Fonction de 2 variables – 1 contrainte inégalité



$$\min_{x_1, x_2} f(x_1, x_2) \text{ sous } c_I(x_1, x_2) \leq 0$$

$$\nabla f(x^*) + \mu \nabla c_I(x^*) = 0$$

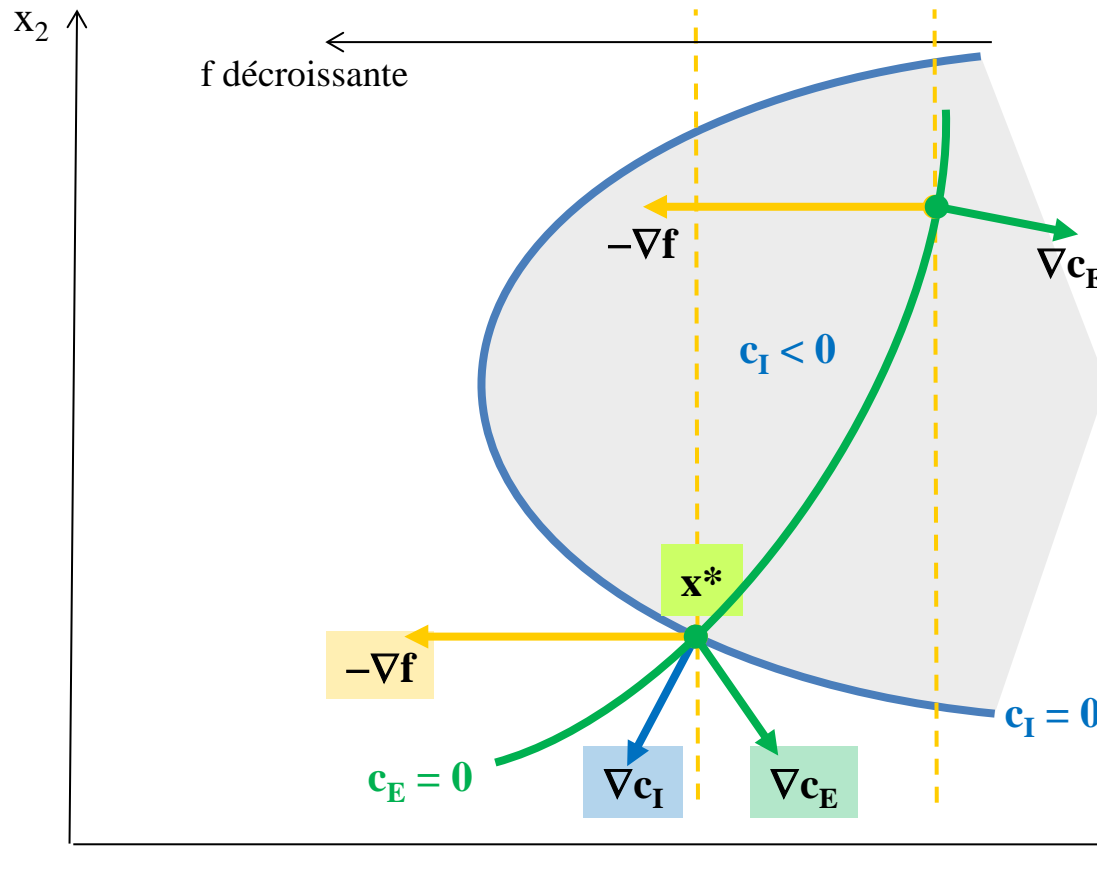
$$\mu \geq 0$$

Sur le schéma :

- $f(x_1, x_2) = x_1$ à minimiser
- $\mu > 0$ ($\mu \approx 2$)

1.4.3 Interprétation géométrique

Fonction de 2 variables – 1 contrainte égalité – 1 contrainte inégalité



$$\min_{x_1, x_2} f(x_1, x_2) \text{ sous } \begin{cases} c_E(x_1, x_2) = 0 \\ c_I(x_1, x_2) \leq 0 \end{cases}$$

$$\nabla f(x^*) + \lambda \nabla c_E(x^*) + \mu \nabla c_I(x^*) = 0$$

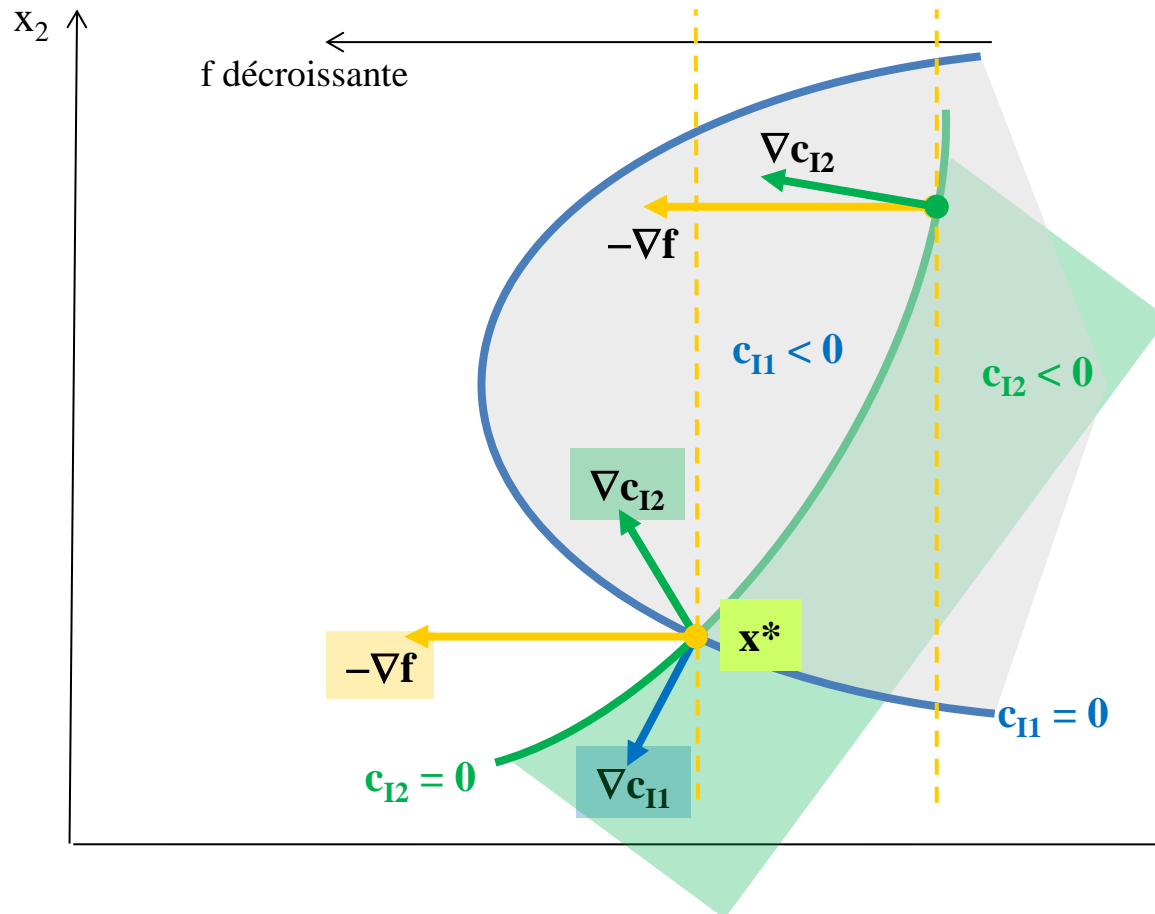
$$\mu \geq 0$$

Sur le schéma :

- $f(x_1, x_2) = x_1$ à minimiser
- $\lambda < 0$ ($\lambda \approx -1.5$)
- $\mu > 0$ ($\mu \approx 1.5$)

1.4.3 Interprétation géométrique

Fonction de 2 variables – 2 contraintes inégalité



$$\min_{x_1, x_2} f(x_1, x_2) \text{ sous } \begin{cases} c_{11}(x_1, x_2) \leq 0 \\ c_{12}(x_1, x_2) \leq 0 \end{cases}$$

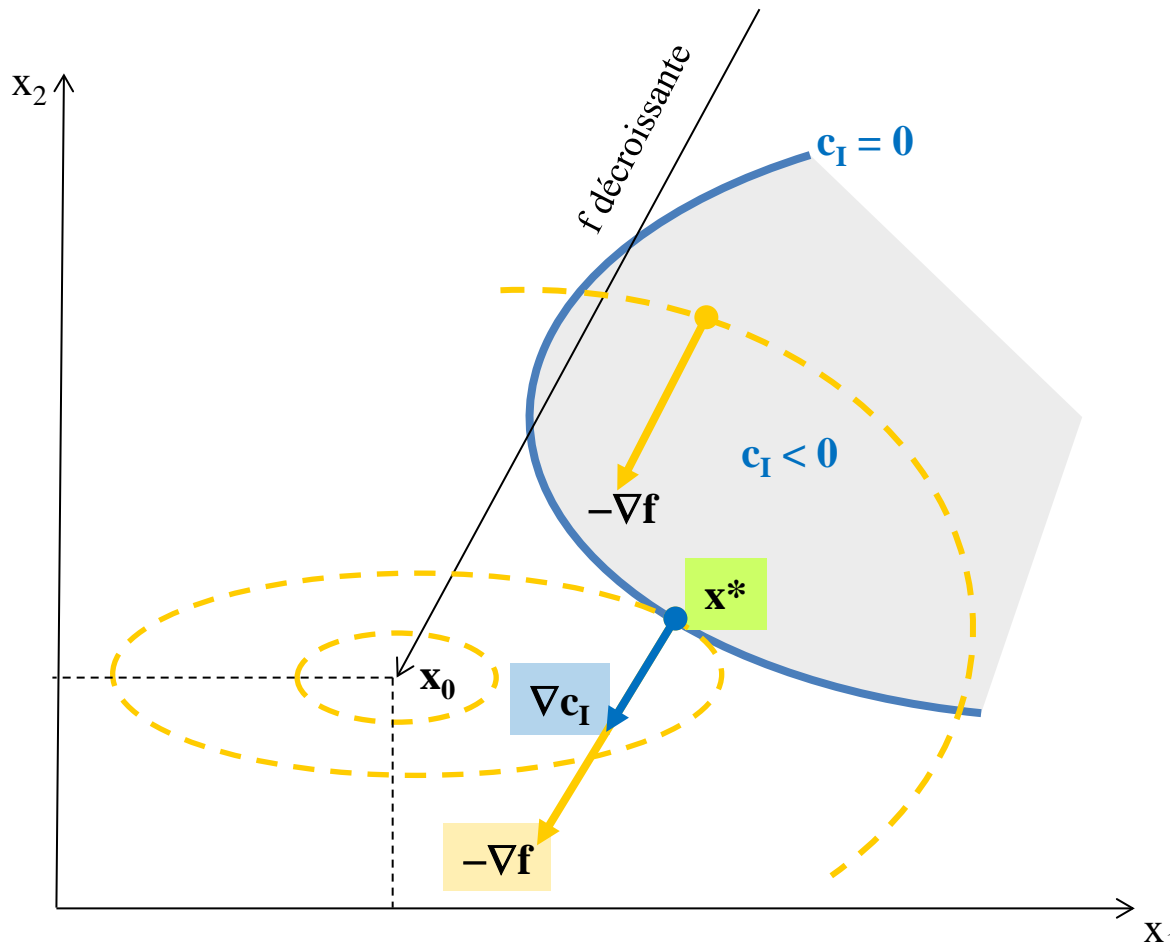
$$\begin{aligned} \nabla f(x^*) + \mu_1 \nabla c_{11}(x^*) + \mu_2 \nabla c_{12}(x^*) &= 0 \\ \mu_1 &\geq 0 \\ \mu_2 &\geq 0 \end{aligned}$$

Sur le schéma :

- $f(x_1, x_2) = x_1$ à minimiser
- $\mu_1 > 0$ ($\mu_1 \approx 1.5$)
- $\mu_2 > 0$ ($\mu_2 \approx 1.5$)

1.4.3 Interprétation géométrique

Fonction de 2 variables – 1 contrainte inégalité



$$\min_{x_1, x_2} f(x_1, x_2) \text{ sous } c_I(x_1, x_2) \leq 0$$

$$\nabla f(x^*) + \mu \nabla c_I(x^*) = 0$$

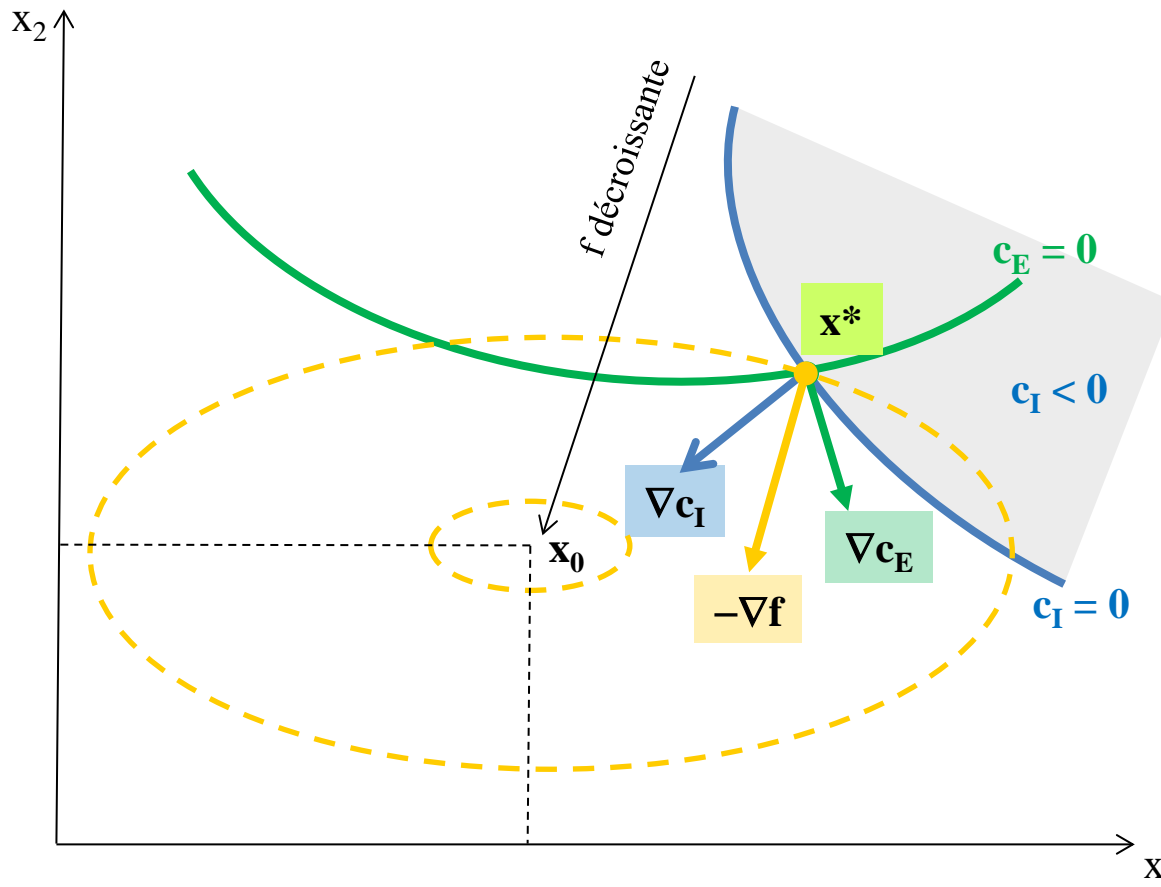
$$\mu \geq 0$$

Sur le schéma :

- $f(x_1, x_2)$ quadratique
- x_0 = minimum sans contrainte
- $\mu > 0$ ($\mu \approx 2$)

1.4.3 Interprétation géométrique

Fonction de 2 variables – 1 contrainte égalité – 1 contrainte inégalité



$$\min_{x_1, x_2} f(x_1, x_2) \text{ sous } \begin{cases} c_E(x_1, x_2) = 0 \\ c_I(x_1, x_2) \leq 0 \end{cases}$$

$$\nabla f(x^*) + \lambda \nabla c_E(x^*) + \mu \nabla c_I(x^*) = 0$$

$$\mu \geq 0$$

Sur le schéma :

- $f(x_1, x_2)$ quadratique
- x_0 = minimum sans contrainte
- $\lambda > 0$ ($\lambda \approx 1.5$)
- $\mu > 0$ ($\mu \approx 1.5$)

1.4.3 Méthode pratique

Problème avec contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \quad \begin{array}{l} \rightarrow p \text{ contraintes d'égalité} \\ \rightarrow q \text{ contraintes d'inégalité} \end{array}$$

La résolution analytique ou numérique nécessite d'**identifier les contraintes actives**.

On se ramène à un problème avec contraintes égalité plus simple.

→ résolution des conditions KKT d'ordre 1

→ vérification des conditions réduites d'ordre 2

Identification des contraintes actives

- Résolution analytique → problème combinatoire (conditions complémentaires)
- Résolution numérique → mise à jour itérative de l'ensemble des contraintes actives

Stratégie itérative d'identification

- On cherche un déplacement à partir du point courant sans tenir compte des contraintes inégalité
 - Le déplacement peut rendre actives certaines contraintes inégalité.
 - On reprend la recherche en ajoutant la première contrainte inégalité activée.
- résolution d'une succession de problèmes avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow m \text{ contraintes actives}$$

1.4.3 Exemple

Problème avec 2 contraintes inégalité

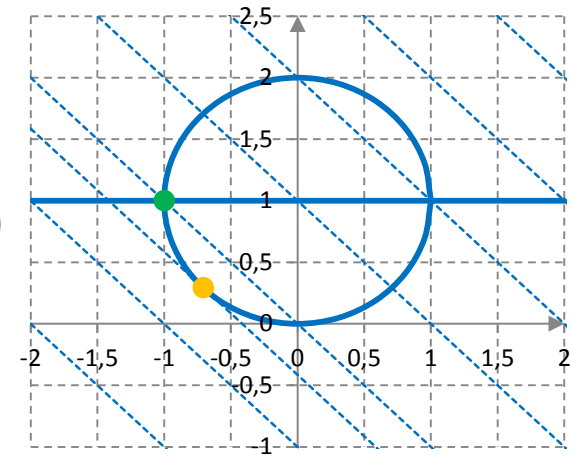
$$\min_{x_1, x_2} f(x) = x_1 + x_2 \quad \text{sous} \quad \begin{cases} c_1(x) = x_1^2 + (x_2 - 1)^2 - 1 \leq 0 \\ c_2(x) = 1 - x_2 \leq 0 \end{cases}$$

- Lagrangien : $L(x, \mu) = f(x) + \mu_1 c_1(x) + \mu_2 c_2(x)$
 $= x_1 + x_2 + \mu_1 (x_1^2 + (x_2 - 1)^2 - 1) + \mu_2 (1 - x_2)$

- Conditions KKT d'ordre 1

$$\begin{cases} 1 + 2\mu_1 x_1 = 0 \\ 1 + 2\mu_1 (x_2 - 1) - \mu_2 = 0 \\ x_1^2 + (x_2 - 1)^2 - 1 \leq 0 \\ 1 - x_2 \leq 0 \\ \mu_1 c_1(x) = 0 \\ \mu_2 c_2(x) = 0 \\ \mu_1, \mu_2 \geq 0 \end{cases}$$

→ conditions complémentaires : **4 combinaisons possibles**



Identification des contraintes actives

Problème combinatoire : il faut essayer les 4 possibilités $\begin{cases} \mu_1 = 0 \text{ ou } c_1(x) = 0 \\ \mu_2 = 0 \text{ ou } c_2(x) = 0 \end{cases}$

1.4.3 Exemple

Problème avec 2 contraintes inégalité

$$\min_{x_1, x_2} f(x) = x_1 + x_2 \quad \text{sous} \quad \begin{cases} c_1(x) = x_1^2 + (x_2 - 1)^2 - 1 \leq 0 \\ c_2(x) = 1 - x_2 \leq 0 \end{cases}$$

- Si $\mu_1 = 0 \rightarrow$ incompatible équation $1 + 2\mu_1 x_1 = 0$

$$\Rightarrow c_1(x) = 0 \rightarrow c_1 \text{ contrainte active}$$

- Si $\mu_2 = 0 \Rightarrow 1 + 2\mu_1(x_2 - 1) = 0$
 \rightarrow incompatible équations $\begin{cases} 1 - x_2 \leq 0 \\ \mu_1 \geq 0 \end{cases}$

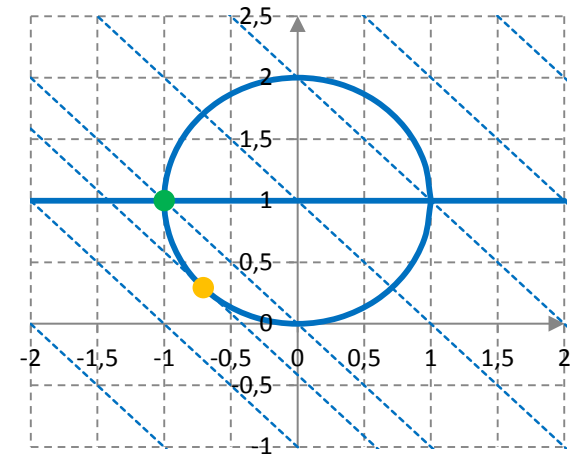
$$\Rightarrow c_2(x) = 0 \rightarrow c_2 \text{ contrainte active}$$

- Combinaison retenue $\begin{cases} c_1(x) = 0 \\ c_2(x) = 0 \end{cases} \Rightarrow \begin{cases} x_1 = \pm 1 \\ x_2 = 1 \end{cases} \Rightarrow \begin{cases} \mu_1 = \mp 0.5 \text{ et } \mu_1 \geq 0 \rightarrow x_1 = -1 \\ \mu_2 = 1 \end{cases}$

- Solution : $\begin{cases} x_1 = -1 \\ x_2 = 1 \\ \mu_1 = 0.5 \\ \mu_2 = 1 \end{cases}$

Vérification condition d'ordre 2 : cône admissible vide
 (2 contraintes actives)

\rightarrow **minimum local**



1.4.3 Exemple

Changement de sens contrainte 2

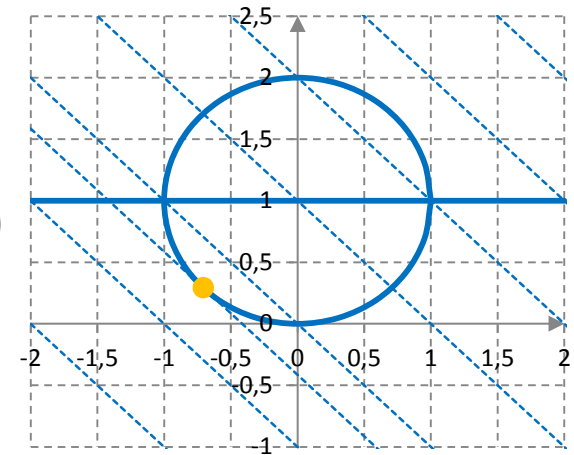
$$\min_{x_1, x_2} f(x) = x_1 + x_2 \quad \text{sous} \quad \begin{cases} c_1(x) = x_1^2 + (x_2 - 1)^2 - 1 \leq 0 \\ c_2(x) = x_2 - 1 \leq 0 \end{cases}$$

- Lagrangien : $L(x, \mu) = f(x) + \mu_1 c_1(x) + \mu_2 c_2(x)$
 $= x_1 + x_2 + \mu_1 (x_1^2 + (x_2 - 1)^2 - 1) + \mu_2 (x_2 - 1)$

- Conditions KKT d'ordre 1

$$\begin{cases} 1 + 2\mu_1 x_1 = 0 \\ 1 + 2\mu_1 (x_2 - 1) + \mu_2 = 0 \\ x_1^2 + (x_2 - 1)^2 - 1 \leq 0 \\ x_2 - 1 \leq 0 \\ \mu_1 c_1(x) = 0 \\ \mu_2 c_2(x) = 0 \\ \mu_1, \mu_2 \geq 0 \end{cases}$$

→ conditions complémentaires : **4 combinaisons possibles**



Identification des contraintes actives

Problème combinatoire : il faut essayer les 4 possibilités $\begin{cases} \mu_1 = 0 \text{ ou } c_1(x) = 0 \\ \mu_2 = 0 \text{ ou } c_2(x) = 0 \end{cases}$

1.4.3 Exemple

Changement de sens contrainte 2

$$\min_{x_1, x_2} f(x) = x_1 + x_2 \quad \text{sous} \quad \begin{cases} c_1(x) = x_1^2 + (x_2 - 1)^2 - 1 \leq 0 \\ c_2(x) = x_2 - 1 \leq 0 \end{cases}$$

- Si $\mu_1 = 0 \rightarrow$ incompatible équation $1 + 2\mu_1 x_1 = 0$

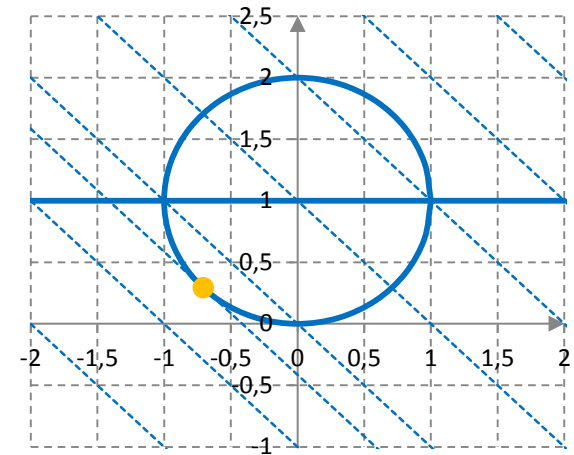
$$\Rightarrow c_1(x) = 0 \rightarrow c_1 \text{ contrainte active}$$

- Si $c_2(x) = 0 \Rightarrow \begin{cases} x_1 = \pm 1 \\ x_2 = 1 \end{cases} \Rightarrow \begin{cases} \mu_1 = \mp 0.5 \\ \mu_2 = -1 \end{cases}$

$$\rightarrow \text{incompatible condition} \quad \mu_2 \geq 0$$

- Si $\mu_2 = 0 \Rightarrow \begin{cases} 1 + 2\mu_1 x_1 = 0 \\ 1 + 2\mu_1 (x_2 - 1) = 0 \\ x_1^2 + (x_2 - 1)^2 - 1 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = -1/(2\mu_1) \\ x_2 = 1 - 1/(2\mu_1) \\ \mu_1 = 1/\sqrt{2} > 0 \end{cases} \Rightarrow \begin{cases} x_1 = -1/\sqrt{2} \\ x_2 = 1 - 1/\sqrt{2} \\ \mu_1 = 1/\sqrt{2} \end{cases}$

- **Vérification condition d'ordre 2 :** $\nabla_{xx}^2 L(x, \mu) = \begin{pmatrix} 2\mu_1 & 0 \\ 0 & 2\mu_1 \end{pmatrix} > 0 \rightarrow \text{minimum local}$
 (1 contrainte active)



1.4.3 Exemple

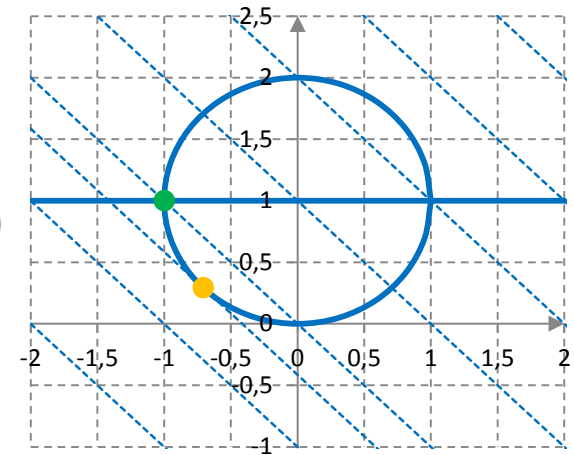
Passage contrainte 1 en égalité

$$\min_{x_1, x_2} f(x) = x_1 + x_2 \quad \text{sous} \quad \begin{cases} c_1(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0 \\ c_2(x) = 1 - x_2 \leq 0 \end{cases}$$

- Lagrangien : $L(x, \mu) = f(x) + \lambda_1 c_1(x) + \mu_2 c_2(x)$
 $= x_1 + x_2 + \lambda_1 (x_1^2 + (x_2 - 1)^2 - 1) + \mu_2 (1 - x_2)$

- Conditions KKT d'ordre 1

$$\begin{cases} 1 + 2\lambda_1 x_1 = 0 \\ 1 + 2\lambda_1 (x_2 - 1) - \mu_2 = 0 \\ x_1^2 + (x_2 - 1)^2 - 1 = 0 \\ 1 - x_2 \leq 0 \\ \mu_2 c_2(x) = 0 \quad \rightarrow \text{conditions complémentaires : 2 combinaisons possibles} \\ \mu_2 \geq 0 \end{cases}$$



Identification des contraintes actives

Problème combinatoire : il faut essayer les 2 possibilités $\mu_2 = 0$ ou $c_2(x) = 0$

1.4.3 Exemple

Passage contrainte 1 en égalité

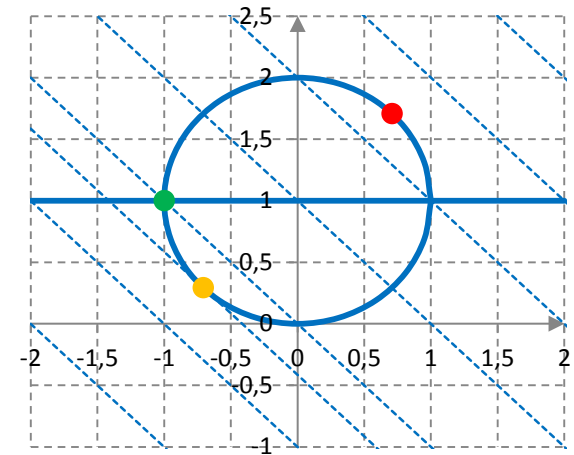
$$\min_{x_1, x_2} f(x) = x_1 + x_2 \quad \text{sous} \quad \begin{cases} c_1(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0 \\ c_2(x) = 1 - x_2 \leq 0 \end{cases}$$

$$\bullet \text{ Si } \mu_2 = 0 \Rightarrow \begin{cases} 1 + 2\lambda_1 x_1 = 0 \\ 1 + 2\lambda_1 (x_2 - 1) = 0 \\ x_1^2 + (x_2 - 1)^2 - 1 = 0 \end{cases} \rightarrow \lambda_1 \neq 0$$

$$\Rightarrow \begin{cases} x_1 = -1/(2\lambda_1) \\ x_2 = 1 - 1/(2\lambda_1) \\ \lambda_1 = \pm 1/\sqrt{2} \end{cases}$$

$$1 - x_2 \leq 0 \Rightarrow \begin{cases} x_1 = 1/\sqrt{2} \\ x_2 = 1 + 1/\sqrt{2} \\ \lambda_1 = -1/\sqrt{2} \end{cases}$$

$$\bullet \text{ Vérification condition d'ordre 2 : } \nabla_{xx}^2 L(x, \mu) = \begin{pmatrix} 2\lambda_1 & 0 \\ 0 & 2\lambda_1 \end{pmatrix} < 0 \rightarrow \text{maximum local} \\ \rightarrow \text{solution rejetée}$$



1.4.3 Exemple

Passage contrainte 1 en égalité

$$\min_{x_1, x_2} f(x) = x_1 + x_2 \quad \text{sous} \quad \begin{cases} c_1(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0 \\ c_2(x) = 1 - x_2 \leq 0 \end{cases}$$

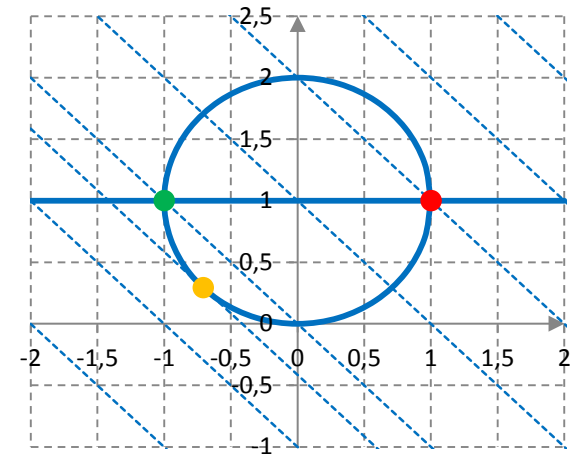
- Si $c_2(x) = 0 \Rightarrow \begin{cases} x_1 = \pm 1 \\ x_2 = 1 \end{cases} \Rightarrow \begin{cases} \lambda_1 = \mp 0.5 \\ \mu_2 = 1 \end{cases}$

- **Vérification condition d'ordre 2**

$$\nabla_{xx}^2 L(x, \mu) = \begin{pmatrix} 2\lambda_1 & 0 \\ 0 & 2\lambda_1 \end{pmatrix}$$

Cône admissible vide : $\nabla c_1 = \begin{pmatrix} 2x_1 \\ 2(x_2 - 1) \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 0 \end{pmatrix}$, $\nabla c_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \rightarrow 2 \text{ contraintes actives}$

\rightarrow **2 minima locaux** : $\begin{cases} x_1 = -1 \\ x_2 = 1 \\ \lambda_1 = 0.5 \\ \mu_2 = 1 \end{cases} \rightarrow f(x) = 0 \quad \text{et} \quad \begin{cases} x_1 = 1 \\ x_2 = 1 \\ \lambda_1 = -0.5 \\ \mu_2 = 1 \end{cases} \rightarrow f(x) = 2$



1.4.3 Méthode pratique

Problème avec contraintes actives

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow \text{ m contraintes actives}$$

Résolution des conditions KKT

On cherche $x^* \in \mathbb{R}^n$ et $\lambda^* \in \mathbb{R}^m$ vérifiant les conditions KKT.

- **Condition nécessaire du 1^{er} ordre**

$$\begin{cases} \nabla_x L(x^*, \lambda^*) = 0 & \rightarrow n \text{ équations} \\ \nabla_\lambda L(x^*, \lambda^*) = 0 & \rightarrow m \text{ équations} \end{cases}$$

Les n équations $\nabla_x L(x^*, \lambda^*)$ permettent d'exprimer $x^* \in \mathbb{R}^n$ en fonction de $\lambda^* \in \mathbb{R}^m$

On remplace ensuite $x^*(\lambda^*)$ dans les m équations $\nabla_\lambda L(x^*, \lambda^*)$.

\rightarrow **système de m équations à m inconnues $\lambda^* \in \mathbb{R}^m$**

- **Condition nécessaire du 2^{ème} ordre**

Il faut vérifier que : $\begin{cases} d^T \nabla_{xx}^2 L(x^*, \lambda^*) d \geq 0 \\ \forall d / d^T \nabla c(x^*) = 0 \end{cases} \quad \rightarrow \text{ hessien du lagrangien semi-défini positif sur le cône admissible}$

Condition difficile à vérifier sous cette forme \rightarrow passage au **hessien réduit**

1.4.3 Méthode pratique

Problème avec contraintes actives

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow \text{ m contraintes actives}$$

Problème équivalent

- Les conditions nécessaires de minimum de f sous contraintes sont :

$$\begin{cases} \nabla_x L(x^*, \lambda^*) = 0 \\ \nabla_\lambda L(x^*, \lambda^*) = 0 \\ d^T \nabla_{xx}^2 L(x^*, \lambda^*) d \geq 0, \forall d / d^T \nabla c(x^*) = 0 \end{cases}$$

- On observe qu'il s'agit également des conditions nécessaires du problème :

$$\min_{x \in \mathbb{R}^n} L(x, \lambda^*) \text{ sous } c(x) = 0$$

- Il est équivalent de minimiser $f(x)$ ou $L(x, \lambda^*)$, **si l'on connaît λ^*** .

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \Leftrightarrow \quad \min_{x \in \mathbb{R}^n} L(x, \lambda^*) \text{ sous } c(x) = 0$$

- On écrit les conditions nécessaires sur le modèle quadratique-linéaire local, puis on applique la technique de réduction des contraintes linéaires.

1.4.3 Méthode pratique

Problème équivalent

$$\min_{x \in \mathbb{R}^n} L(x, \lambda^*) \text{ sous } c(x) = 0$$

Modèle quadratique-linéaire

- Modèle quadratique du critère : $\hat{L}(x^* + p) = L(x^*, \lambda^*) + p^T \nabla_x L(x^*, \lambda^*) + \frac{1}{2} p^T \nabla_{xx}^2 L(x^*, \lambda^*) p$

En notant : $\begin{cases} g_L(x^*) = \nabla_x L(x^*, \lambda^*) \\ H_L(x^*) = \nabla_{xx}^2 L(x^*, \lambda^*) \end{cases} \rightarrow \begin{array}{l} \text{gradient du lagrangien par rapport à } x \\ \text{hessien du lagrangien par rapport à } x \end{array}$

$$\rightarrow \hat{L}(x^* + p) = L(x^*, \lambda^*) + p^T g_L(x^*) + \frac{1}{2} p^T H_L(x^*) p$$

- Modèle linéaire des contraintes : $\hat{c}(x^* + p) = c(x^*) + \nabla c(x^*)^T p$ avec $c(x^*) = 0$

En notant : $\begin{cases} A = \nabla c(x^*)^T \\ p = Y p_Y + Z p_Z \end{cases} \text{ avec } \begin{array}{l} AY \text{ inversible} \\ AZ = 0 \text{ (espace nul)} \end{array}$

$$\nabla c(x^*)^T p = 0 \Leftrightarrow \begin{cases} AY p_Y = 0 \\ p_Z \text{ libre} \end{cases} \Leftrightarrow \begin{cases} p_Y = 0 \\ p_Z \text{ libre} \end{cases} \text{ car } AY \text{ inversible}$$

- Problème réduit : $\min_{p \in \mathbb{R}^n} \hat{L}(x^* + p) \text{ sous } \hat{c}(x^* + p) = 0 \Leftrightarrow \min_{p_Z \in \mathbb{R}^{n-m}} \hat{L}(x^* + Z p_Z)$

1.4.3 Méthode pratique

Problème réduit

$\min_{p_Z \in \mathbb{R}^{n-m}} \hat{L}(x^* + Zp_Z) \rightarrow$ problème sans contrainte à $n-m$ variables p_Z

avec $\hat{L}(x^* + Zp_Z) = L(x^*, \lambda^*) + p_Z^T Z^T g_L(x^*) + \frac{1}{2} p_Z^T Z^T H_L(x^*) Z p_Z$

Conditions nécessaires de minimum du problème réduit

$$\hat{L}(x^* + Zp_Z) \geq \hat{L}(x^*), \forall p_Z \Rightarrow p_Z^T Z^T g_L(x^*) + \frac{1}{2} p_Z^T Z^T H_L(x^*) Z p_Z \geq 0, \forall p_Z \Rightarrow \begin{cases} Z^T g_L(x^*) = 0 \\ Z^T H_L(x^*) Z \geq 0 \end{cases}$$

- Condition réduite d'ordre 1 : $Z^T g_L(x^*) = 0 \Leftrightarrow Z^T \nabla_x L(x^*, \lambda^*) = Z^T (\nabla f(x^*) + \nabla c(x^*) \lambda^{*T}) = 0$
 $\Leftrightarrow Z^T \nabla f(x^*) = 0$ car $\nabla c(x^*)^T Z = 0$
- Condition réduite d'ordre 2 : $Z^T H_L(x^*) Z \geq 0$
- $\begin{cases} g_Z = Z^T g & \rightarrow \text{gradient réduit du critère} & g(x) = \nabla f(x) \\ H_Z = Z^T H_L Z & \rightarrow \text{hessien réduit du lagrangien} & H_L(x) = \nabla_{xx}^2 L(x, \lambda^*) \end{cases}$

- x^* minimum local $\Rightarrow \begin{cases} g_Z(x^*) = 0 & \rightarrow \text{gradient réduit du critère nul} \\ H_Z(x^*) \geq 0 & \rightarrow \text{hessien réduit du lagrangien semi-défini positif} \end{cases}$

1.4.3 Exemple

Problème de la boîte

- Réaliser une boîte cylindrique de volume donné V_0 et de surface minimale
- Dimensions : hauteur = h , rayon = r

Formulation du problème

- Surface : $S = 2\pi r^2 + 2\pi rh$
 - Volume : $V = \pi r^2 h$
- $$\rightarrow \min_{h,r} S(h,r) \text{ sous } V(h,r) = V_0$$

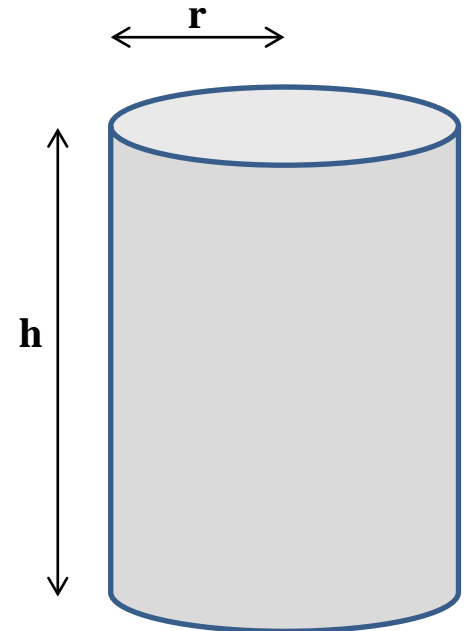
Résolution

On note : $V_0 = 2\pi v_0$

- Lagrangien : $L(h,r,\lambda) = 2\pi r^2 + 2\pi rh + \lambda(\pi r^2 h - 2\pi v_0)$
- Conditions KKT

$$\begin{cases} 2\pi r + \lambda \pi r^2 = 0 \\ 4\pi r + 2\pi h + 2\lambda \pi rh = 0 \\ \pi r^2 h - 2\pi v_0 = 0 \end{cases} \Rightarrow \begin{cases} \lambda r + 2 = 0 \\ 2r + h + \lambda rh = 0 \\ r^2 h - 2v_0 = 0 \end{cases} \Rightarrow \begin{cases} \lambda r = -2 \\ h = 2r \\ r^3 = v_0 \end{cases}$$

- Solution : $\begin{cases} r = v_0^{1/3} \\ h = 2v_0^{1/3} \end{cases} \Rightarrow S = 6\pi v_0^{2/3} = 3(2\pi)^{1/3} V_0^{2/3}$



1.4.3 Exemple

Vérification des conditions réduites

Il faut choisir une base de réduction, puis vérifier les conditions réduites de minimum local

$$\begin{cases} g_Z(x^*) = Z^T \nabla f(x^*) = 0 & \rightarrow \text{gradient réduit du critère nul} \\ H_Z(x^*) = Z^T \nabla_{xx}^2 L(x, \lambda^*) Z \geq 0 & \rightarrow \text{hessien réduit du lagrangien semi-défini positif} \end{cases}$$

- Gradient du critère : $g(h, r) = \nabla_{h,r} f(h, r) = 2\pi \begin{pmatrix} r \\ 2r + h \end{pmatrix}$
 - Hessien du lagrangien : $L(h, r, \lambda) = 2\pi r^2 + 2\pi r h + \lambda(\pi r^2 h - 2\pi v_0)$
- $$\Rightarrow g_L(h, r) = \nabla_{h,r} L(h, r, \lambda) = \pi \begin{pmatrix} 2r + \lambda r^2 \\ 4r + 2h + 2\lambda r h \end{pmatrix}, \quad H_L(h, r) = \nabla_{h,r}^2 L(h, r, \lambda) = 2\pi \begin{pmatrix} 0 & 1 + \lambda r \\ 1 + \lambda r & 2 + \lambda h \end{pmatrix}$$

Choix d'une base de réduction

- Contrainte : $c(h, r) = \pi r^2 h - 2\pi v_0 = 0 \Rightarrow \nabla c^T = \begin{pmatrix} \pi r^2 & 2\pi r h \end{pmatrix}$
- **Choix de la base avec la variable h** : $A = \nabla c^T = \begin{pmatrix} \pi r^2 & 2\pi r h \end{pmatrix} = \begin{pmatrix} h & r \\ B & N \end{pmatrix}$

- Base de l'espace nul : $Z = \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix} = \begin{pmatrix} -2\pi r h / \pi r^2 \\ 1 \end{pmatrix} = \begin{pmatrix} -2h / r \\ 1 \end{pmatrix}$

1.4.3 Exemple

Vérification des conditions réduites

- **Gradient réduit du critère :** $g_Z(h, r) = Z^T g(h, r) = 2\pi \begin{pmatrix} -2h/r \\ 1 \end{pmatrix}^T \begin{pmatrix} r \\ 2r+h \end{pmatrix} = 2\pi(2r-h)$

On vérifie que le gradient réduit est nul : $h = 2r \Rightarrow g_Z(h, r) = 0$

- **Hessien réduit du lagrangien :** $H_Z(h, r) = Z^T H(h, r) Z = 2\pi \begin{pmatrix} -2h/r \\ 1 \end{pmatrix}^T \begin{pmatrix} 0 & 1+\lambda r \\ 1+\lambda r & 2+\lambda r \end{pmatrix} \begin{pmatrix} -2h/r \\ 1 \end{pmatrix}$
 $\Rightarrow H_Z(h, r) = 2\pi \left(2 - 4\frac{h}{r} - 3\lambda h \right)$

On vérifie que le hessien réduit est semi-défini positif

$$\begin{cases} \lambda r = -2 \\ h = 2r \end{cases} \Rightarrow H_Z(h, r) = 2\pi \left(2 - \frac{h}{r} (4 + 3\lambda r) \right) = 12\pi > 0$$

1.4.3 Exemple

Résolution par élimination

- Contrainte : $c(h, r) = \pi r^2 h - 2\pi v_0 = 0 \Rightarrow h = \frac{2v_0}{r^2}$
- Elimination de la variable h : $S(h, r) = 2\pi r^2 + 2\pi r h \Rightarrow S(r) = 2\pi r^2 + \frac{4\pi v_0}{r}$
- Gradient : $\frac{dS}{dr}(r) = 4\pi r - \frac{4\pi v_0}{r^2} = 4\pi r \left(1 - \frac{v_0}{r^3}\right)$
- Hessien : $\frac{d^2S}{dr^2}(r) = 4\pi + \frac{8\pi v_0}{r^3} = 4\pi \left(1 + 2\frac{v_0}{r^3}\right)$
- Minimum de $S(r)$: $\begin{cases} \frac{dS}{dr}(r) = 0 \\ \frac{d^2S}{dr^2}(r) \geq 0 \end{cases} \Rightarrow \begin{cases} r^3 = v_0 \\ \frac{d^2S}{dr^2}(r) = 12\pi > 0 \end{cases}$

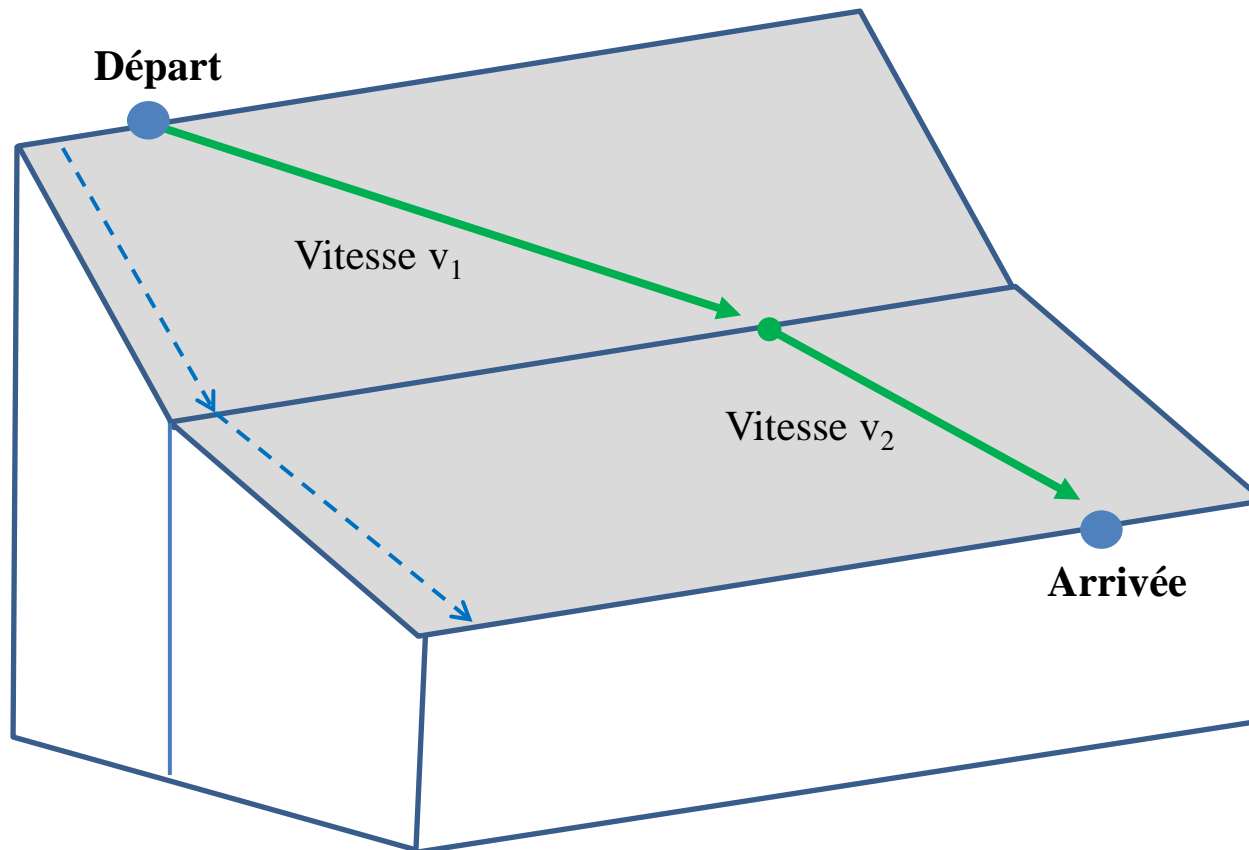
Lien avec les conditions réduites

- Gradient réduit
 $g_Z(h, r) = 2\pi(2r - h)$ avec $h = \frac{2v_0}{r^2} \Rightarrow g_Z(h, r) = 2\pi \left(2r - \frac{2v_0}{r^2}\right) = 4\pi r \left(1 - \frac{v_0}{r^3}\right) = \frac{dS}{dr}(r)$
- Hessien réduit
 \rightarrow pas de relation directe entre H_Z et $\frac{d^2S}{dr^2}$ (contrainte non linéaire)

1.4.3 Exemples

Problème du skieur

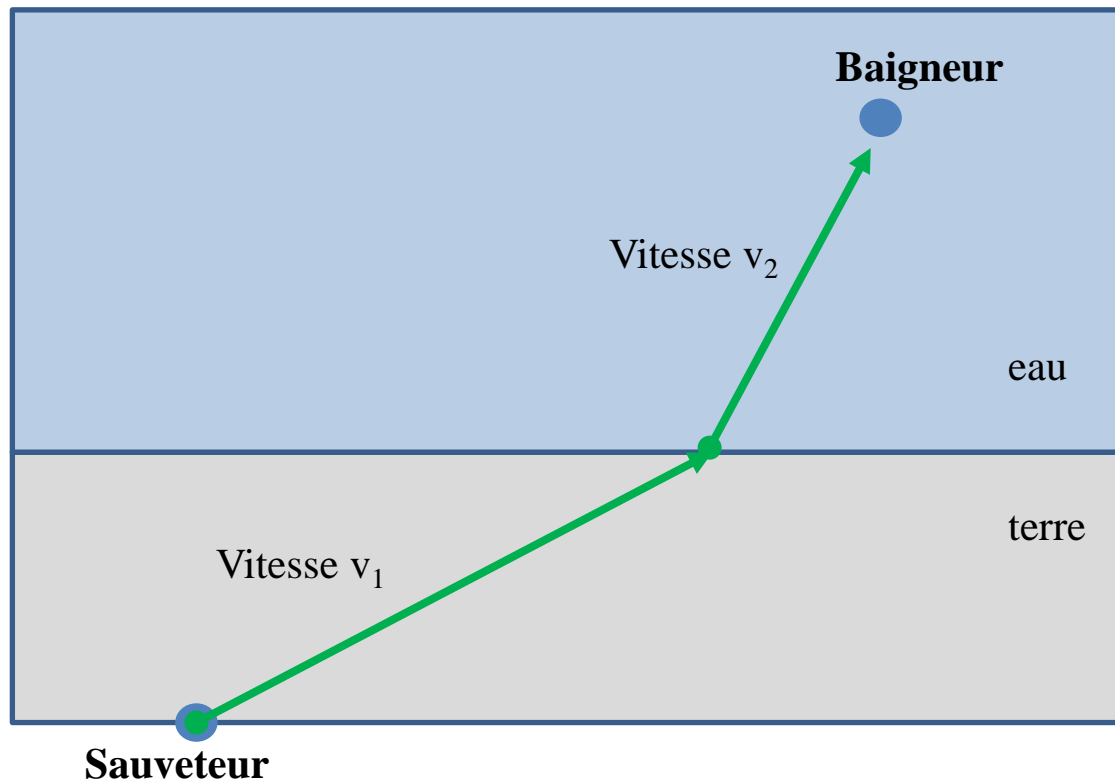
- Descendre du départ à l'arrivée le plus vite possible
- 2 zones de pentes différentes : vitesse v_1 , puis v_2



1.4.3 Exemples

Problème du sauveteur

- Aller secourir le baigneur qui se noie le plus vite possible
- Course sur terre, puis nage dans l'eau : vitesse v_1 , puis v_2



1.4.3 Exemples

Problème du sauveteur

- Données du problème : l_0, l_1, l_2, v_1, v_2

Distance sur terre : $d_1 = \frac{l_1}{\cos \theta_1}$

Durée de course : $t_1 = \frac{d_1}{v_1}$

Distance dans l'eau : $d_2 = \frac{l_2}{\cos \theta_2}$

Durée de nage : $t_2 = \frac{d_2}{v_2}$

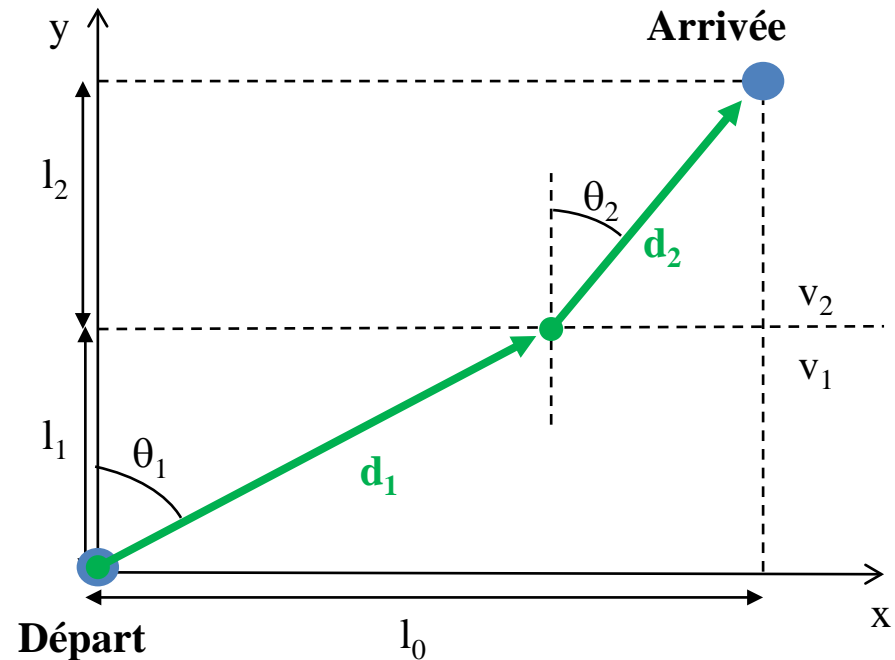
Distance suivant x : $L = d_1 \sin \theta_1 + d_2 \sin \theta_2$

- Formulation du problème

Variables : θ_1, θ_2

Contrainte : $L = l_0 \rightarrow$ atteindre le point visé

Critère : $T = t_1 + t_2 \rightarrow$ durée totale à minimiser



1.4.3 Exemples

Problème du sauveteur

- Formulation du problème

$$\min_{\theta_1, \theta_2} T(\theta_1, \theta_2) \text{ sous } L(\theta_1, \theta_2) = l_0$$

$$\Leftrightarrow \min_{\theta_1, \theta_2} T = \frac{l_1}{v_1 \cos \theta_1} + \frac{l_2}{v_2 \cos \theta_2} \text{ sous } L = l_1 \tan \theta_1 + l_2 \tan \theta_2 = l_0$$

- Résolution du problème

$$\text{Lagrangien : } L(\theta_1, \theta_2, \lambda) = \frac{l_1}{v_1 \cos \theta_1} + \frac{l_2}{v_2 \cos \theta_2} + \lambda(l_1 \tan \theta_1 + l_2 \tan \theta_2 - l_0)$$

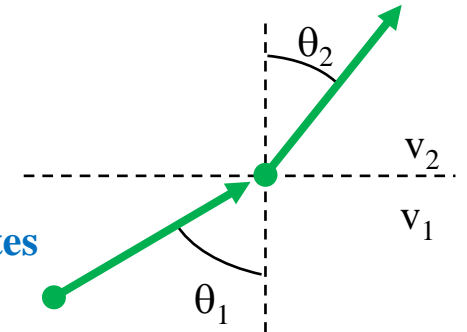
$$\text{Conditions KKT : } \begin{cases} \frac{l_1 \sin \theta_1}{v_1 \cos^2 \theta_1} + \lambda l_1 \frac{1}{\cos^2 \theta_1} = 0 \\ \frac{l_2 \sin \theta_2}{v_2 \cos^2 \theta_2} + \lambda l_2 \frac{1}{\cos^2 \theta_2} = 0 \\ l_1 \tan \theta_1 + l_2 \tan \theta_2 = l_0 \end{cases} \Rightarrow \begin{cases} \sin \theta_1 + \lambda v_1 = 0 \\ \sin \theta_2 + \lambda v_2 = 0 \\ l_1 \tan \theta_1 + l_2 \tan \theta_2 = l_0 \end{cases}$$

1.4.3 Exemples

Problème du sauveteur

- Conditions KKT :
$$\begin{cases} \sin \theta_1 + \lambda v_1 = 0 \\ \sin \theta_2 + \lambda v_2 = 0 \\ l_1 \tan \theta_1 + l_2 \tan \theta_2 = l_0 \end{cases}$$

- θ_1, θ_2 vérifient : $\frac{\sin \theta_1}{\sin \theta_2} = \frac{v_1}{v_2} \rightarrow$ **loi de la réfraction de Descartes**



- Pour résoudre complètement

On exprime θ_1, θ_2 en fonction de λ :

$$\begin{cases} \sin \theta_1 = -\lambda v_1 \Rightarrow \cos \theta_1 = \sqrt{1 - \lambda^2 v_1^2} \\ \sin \theta_2 = -\lambda v_2 \Rightarrow \cos \theta_2 = \sqrt{1 - \lambda^2 v_2^2} \end{cases}$$

On remplace dans la contrainte :

$$l_1 \tan \theta_1 + l_2 \tan \theta_2 = l_0 \Rightarrow \frac{\lambda l_1 v_1}{\sqrt{1 - \lambda^2 v_1^2}} + \frac{\lambda l_2 v_2}{\sqrt{1 - \lambda^2 v_2^2}} = -l_0$$

On obtient une équation en λ :

\rightarrow équation de degré 4

\rightarrow solution $\lambda^* \rightarrow \theta_1^*, \theta_2^*$

1.4.3 Sensibilité aux contraintes

Problème avec contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$$

→ p contraintes d'égalité
 → q contraintes d'inégalité

$$\Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

→ m contraintes d'égalité (= contraintes actives)

Problème initial

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

- Multiplicateurs : λ^* , $\nabla_x L(x^*, \lambda^*) = 0$
- Solution : x^* , $f(x^*) = f^*$
 $c(x^*) = 0$

Problème perturbé

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = \delta c$$

- **Variation des niveaux de contrainte δc** → **Variation de la solution δx , du coût optimal δf**
- Solution : $x^* + \delta x$, $f(x^* + \delta x) = f^* + \delta f$, $\delta f \in \mathbb{R}$
 $c(x^* + \delta x) = 0 + \delta c$, $\delta c \in \mathbb{R}^m$

1.4.3 Sensibilité aux contraintes

Variation du coût optimal

- Variation du coût et des contraintes à l'ordre 1

$$\begin{cases} f(x^* + \delta x) = f(x^*) + \nabla f(x^*)^T \delta x = f(x^*) + \delta f \\ c(x^* + \delta x) = c(x^*) + \nabla c(x^*)^T \delta x = c(x^*) + \delta c \end{cases} \Rightarrow \begin{cases} \delta f = \nabla f(x^*)^T \delta x \\ \delta c = \nabla c(x^*)^T \delta x \end{cases}$$

- Condition d'optimalité d'ordre 1 du problème initial

$$\nabla_x L(x^*, \lambda^*) = 0 \Rightarrow \nabla f(x^*) + \nabla c(x^*) \lambda^* = 0$$

- Relation entre δf et δc

$$\delta f + \lambda^{*T} \delta c = (\nabla f(x^*)^T + \lambda^{*T} \nabla c(x^*)^T) \delta x = (\nabla f(x^*) + \nabla c(x^*) \lambda^*)^T \delta x = 0$$

$$\Rightarrow \boxed{\delta f = -\lambda^{*T} \delta c = -\sum_{j=1}^m \lambda_j^* \delta c_j}$$

- Une **variation δc_j** du niveau de la contrainte j entraîne une **variation $-\lambda_j^* \delta c_j$** du coût optimal.

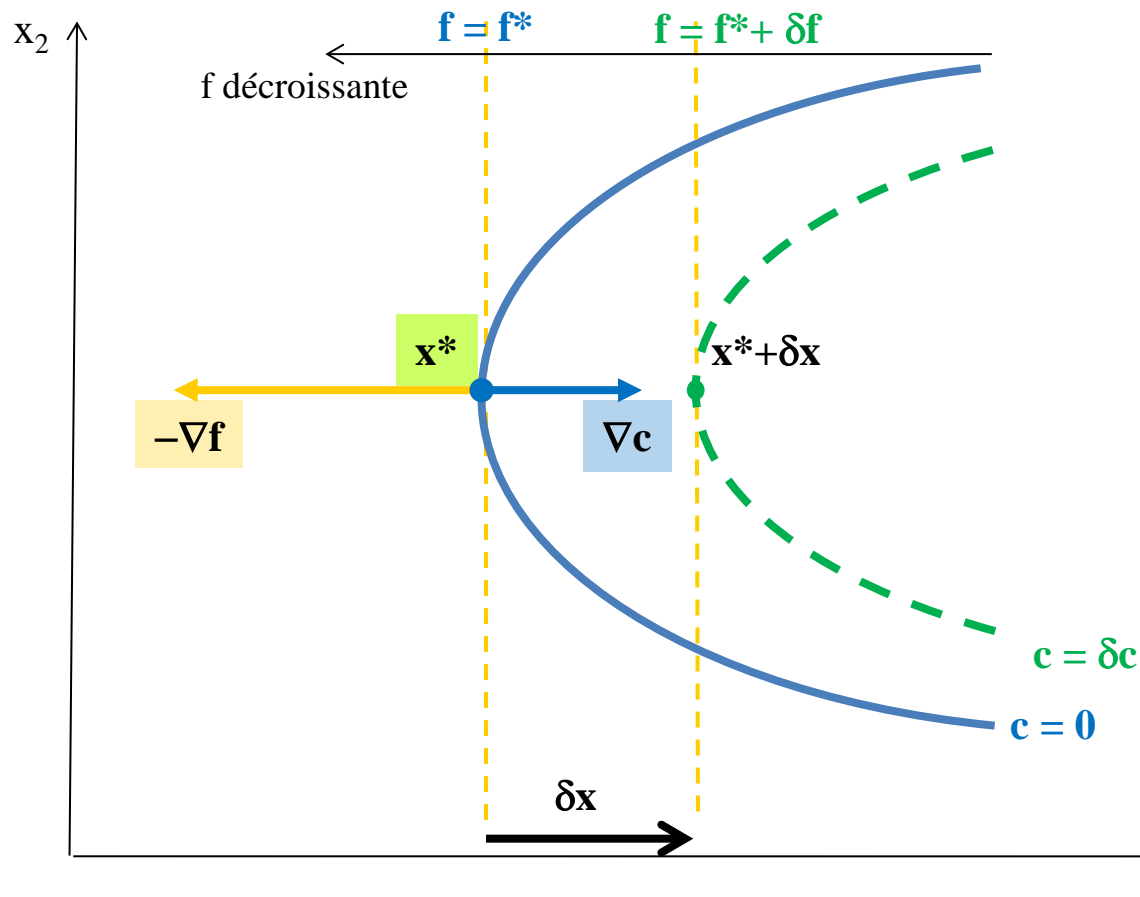
Interprétation

Le multiplicateur donne la **sensibilité** du coût optimal au niveau de la contrainte (au signe près)

→ Convention possible de définition du lagrangien : $L = f + \lambda^T c$
 ou $L = f - \lambda^T c$

1.4.3 Sensibilité aux contraintes

Fonction de 2 variables – 1 contrainte égalité



$$\min_{x_1, x_2} f(x_1, x_2) \text{ sous } c(x_1, x_2) = 0$$

$$\nabla f(x^*) + \lambda \nabla c(x^*) = 0$$

Variations :

- $\delta c = \nabla c \cdot \delta x$
- $\delta f = \nabla f \cdot \delta x = -\lambda \delta c$

Sur le schéma :

- $f(x_1, x_2) = x_1$ à minimiser
- $\lambda < 0$ ($\lambda \approx -2$)
- $\delta c > 0 \rightarrow \delta x > 0$
 $\rightarrow \delta f < 0$

1.4.3 Exemple

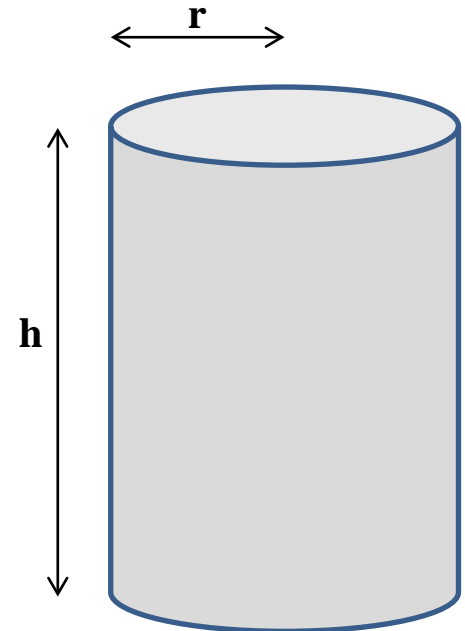
Problème de la boîte

$$\min_{h,r} S(h,r) \text{ sous } V(h,r) = V_0 \text{ avec } \begin{cases} S = 2\pi r^2 + 2\pi rh \\ V = \pi r^2 h \end{cases}$$

→ contrainte en volume de niveau V_0

Solution

$$\begin{cases} r = \left(\frac{V_0}{2\pi} \right)^{\frac{1}{3}} \\ h = 2 \left(\frac{V_0}{2\pi} \right)^{\frac{1}{3}} \\ \lambda = -2 \left(\frac{2\pi}{V_0} \right)^{\frac{1}{3}} \end{cases} \Rightarrow S = 3(2\pi)^{\frac{1}{3}} V_0^{\frac{2}{3}}$$



Sensibilité au niveau de contrainte

$$\frac{dS}{dV_0} = 2(2\pi)^{\frac{1}{3}} V_0^{-\frac{1}{3}} = 2 \left(\frac{2\pi}{V_0} \right)^{\frac{1}{3}} = -\lambda$$

1.4.3 Sensibilité aux paramètres

Problème avec paramètres de modèle

$$\min_{x \in \mathbb{R}^n} f(x, p) \text{ sous } c(x, p) = 0$$

→ m contraintes d'égalité (= contraintes actives)
 r **paramètres de modèle** $p \in \mathbb{R}^r$ (valeurs fixées)

Problème initial

$$\min_{x \in \mathbb{R}^n} f(x, p) \text{ sous } c(x, p) = 0$$

- Multiplicateurs : λ^* , $\nabla_x L(x^*, \lambda^*, p) = 0$
- Solution : x^* , $f(x^*, p) = f^*$
 $c(x^*, p) = 0$

Problème perturbé

$$\min_{x \in \mathbb{R}^n} f(x, p + \delta p) \text{ sous } c(x, p + \delta p) = 0$$

- **Variation des paramètres δp → Variation de la solution δx , du coût optimal δf**
- Solution : $x^* + \delta x$, $f(x^* + \delta x, p + \delta p) = f^* + \delta f$, $\delta f \in \mathbb{R}$
 $c(x^* + \delta x, p + \delta p) = 0$ (même niveau de contrainte = 0)

1.4.3 Sensibilité aux paramètres

Variation du coût optimal

- Variation du coût et des contraintes à l'ordre 1

$$\begin{cases} f(x^* + \delta x, p + \delta p) = f(x^*, p) + \nabla_x f(x^*, p)^T \delta x + \nabla_p f(x^*, p)^T \delta p = f(x^*, p) + \delta f \\ c(x^* + \delta x, p + \delta p) = c(x^*, p) + \nabla_x c(x^*, p)^T \delta x + \nabla_p c(x^*, p)^T \delta p = c(x^*, p) = 0 \end{cases}$$

$$\Rightarrow \begin{cases} \nabla_x f(x^*, p)^T \delta x + \nabla_p f(x^*, p)^T \delta p = \delta f \\ \nabla_x c(x^*, p)^T \delta x + \nabla_p c(x^*, p)^T \delta p = 0 \end{cases}$$

- Condition d'optimalité d'ordre 1 du problème initial

$$\nabla_x L(x^*, \lambda^*, p) = 0 \Rightarrow \nabla_x f(x^*, p) + \nabla_x c(x^*, p) \lambda^* = 0$$

- Relation entre δf et δp

$$\delta f = (\nabla_x f(x^*, p)^T + \lambda^{*T} \nabla_x c(x^*, p)^T) \delta x + (\nabla_p f(x^*, p)^T + \lambda^{*T} \nabla_p c(x^*, p)^T) \delta p$$

$$\Rightarrow \delta f = (\nabla_p f(x^*, p) + \nabla_p c(x^*, p) \lambda^*)^T \delta p = \nabla_p L(x^*, \lambda^*, p)^T \delta p \Rightarrow \frac{df(x^*, p)}{dp} = \nabla_p L(x^*, \lambda^*, p)$$

- Une **variation** δp_j du paramètre j entraîne une **variation** $\frac{\partial L(x^*, \lambda^*, p)}{\partial p_j} \delta p_j$ du coût optimal

→ **sensibilité** du coût aux paramètres de modèle

1.4.3 Exemple

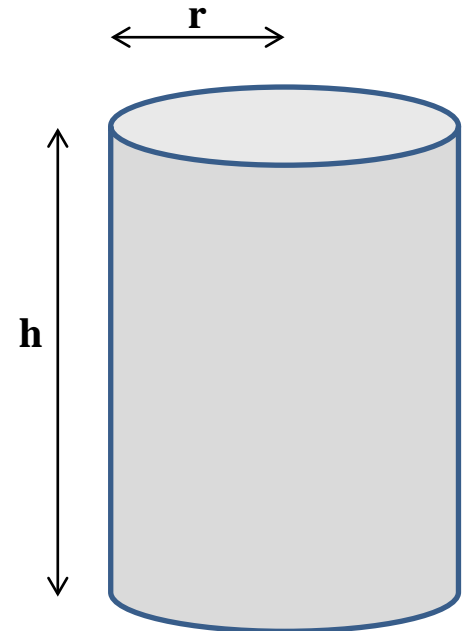
Problème de la boîte

$$\min_{h,r} S(h,r) \text{ sous } M(h,r) = M_0 = \rho V_0 \text{ avec } \begin{cases} S = 2\pi r^2 + 2\pi rh \\ M = \rho V = \pi r^2 h \rho \end{cases}$$

→ contrainte en masse M_0 au lieu de volume V_0 , avec densité ρ

Solution

$$\begin{cases} r = \left(\frac{M_0}{2\pi\rho} \right)^{\frac{1}{3}} \\ h = 2 \left(\frac{M_0}{2\pi\rho} \right)^{\frac{1}{3}} \\ \lambda = -\frac{2}{\rho} \left(\frac{2\pi\rho}{M_0} \right)^{\frac{1}{3}} \end{cases} \Rightarrow \begin{aligned} S &= 3(2\pi)^{\frac{1}{3}} M_0^{\frac{2}{3}} \rho^{-\frac{2}{3}} \\ L &= S + \lambda(M - M_0) \\ \frac{\partial L}{\partial \rho} &= \lambda \frac{\partial M}{\partial \rho} = \lambda \pi r^2 h \\ \frac{\partial L}{\partial \rho} &= -\frac{4\pi}{\rho} \left(\frac{M_0}{2\pi\rho} \right)^{\frac{2}{3}} \end{aligned}$$



Sensibilité au paramètre ρ

$$\frac{dS}{d\rho} = -2(2\pi)^{\frac{1}{3}} M_0^{\frac{2}{3}} \rho^{-\frac{5}{3}} = -\frac{4\pi}{\rho} \left(\frac{M_0}{2\pi\rho} \right)^{\frac{2}{3}} = \frac{\partial L}{\partial \rho}$$

1.4.4 Problème linéaire

- ☐ Forme standard
- ☐ Conditions nécessaires d'optimalité
- ☐ Coûts réduits

1.4.4 Problème linéaire

Problème linéaire

$$\min_{x \in \mathbb{R}^n} c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$$

→ problème linéaire sous forme standard (PL)

Conditions nécessaires d'optimalité à partir du lagrangien

$$\text{Lagrangien : } L(x, \lambda, s) = c^T x + \lambda^T (b - Ax) - s^T x \Rightarrow \begin{cases} \nabla_x L(x, \lambda, s) = c - A^T \lambda - s \\ \nabla_{xx}^2 L(x, \lambda, s) = 0 \end{cases}$$

(x, λ, s) minimum local de (PL)

⇓

- Condition nécessaire d'ordre 1 : $\begin{cases} c - A^T \lambda - s = 0 \\ s \geq 0 \end{cases} \rightarrow \text{contraintes du problème dual}$
- Condition nécessaire d'ordre 2 : $\nabla_{xx}^2 L(x, \lambda, s) \geq 0 \rightarrow \text{vérifiée}$
- Condition complémentaire : $s_i x_i = 0, i = 1, \dots, n$

1.4.4 Problème linéaire

Problème linéaire

$$\min_{x \in \mathbb{R}^n} c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$$

→ problème linéaire sous forme standard (PL)

Conditions nécessaires d'optimalité à partir des dérivées directionnelles

x minimum local de (PL)

⇓

Pour toute direction admissible d : $\nabla f(x)^T d \geq 0 \Rightarrow c^T d \geq 0$

- Toute direction admissible d est combinaison linéaire des directions de base d_j . (contraintes linéaires)

$$d_j = E \begin{pmatrix} d_{jB} \\ d_{jN} \end{pmatrix} \text{ avec } \begin{cases} d_{jB} = -B^{-1}A_j \\ E^T e_j = \begin{pmatrix} 0 \\ d_{jN} \end{pmatrix} \end{cases} \Rightarrow c^T d_j = c_B^T d_{jB} + c_N^T d_{jN} = -c_B^T B^{-1} A_j + c_j$$

- Il suffit de vérifier : $\boxed{c^T d_j \geq 0}$

1.4.4 Problème linéaire

Coûts réduits

x solution de base admissible : $x = E \begin{pmatrix} x_B \\ x_N \end{pmatrix} = E \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} \begin{matrix} \rightarrow m \\ \rightarrow n-m \end{matrix}$

Le **coût réduit** associé à la variable hors base x_j est défini par : $\boxed{\bar{c}_j = c^T d_j = c_j - c_B^T B^{-1} A_j}$

= **dérivée directionnelle** de f suivant la $j^{\text{ème}}$ direction de base pour une variable hors base
 = 0 par extension pour une variable de base

$$AE = (B \ N) \Rightarrow B^{-1}AE = (I \ B^{-1}N) \Rightarrow c_B^T B^{-1}AE = (c_B^T \ c_B^T B^{-1}N) \Rightarrow \bar{c} = (0 \ \bar{c}_N^T)$$

Conditions nécessaires d'optimalité

x^* solution de base non dégénérée

$$x^* \text{ solution de PL} \Rightarrow \bar{c} \geq 0$$

Conditions suffisantes d'optimalité

x^* solution de base admissible

$$\boxed{\bar{c} \geq 0 \Rightarrow x^* \text{ solution de PL}}$$

1.4.4 Problème linéaire

Lien entre multiplicateurs et coûts réduits

- Problème linéaire sous forme standard : $\min_{x \in \mathbb{R}^n} c^T x$ sous $\begin{cases} Ax = b \\ x \geq 0 \end{cases}$
- Lagrangien : $L(x, \lambda, s) = c^T x + \lambda^T (b - Ax) - s^T x \Rightarrow \nabla_x L(x, \lambda, s) = c - A^T \lambda - s$
- Conditions d'ordre 1 : $\begin{cases} A^T \lambda + s = c \\ s \geq 0, s_i x_i = 0, i = 1, \dots, n \end{cases}$
- Base B : $AE = (B \ N) \Rightarrow c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}, s = \begin{pmatrix} s_B \\ s_N \end{pmatrix}, x = \begin{pmatrix} x_B = B^{-1}b \\ x_N = 0 \end{pmatrix}$

$$A^T \lambda + s = c \Leftrightarrow \begin{pmatrix} B^T \\ N^T \end{pmatrix} \lambda + \begin{pmatrix} s_B \\ s_N \end{pmatrix} = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$$

$$s \geq 0, s_i x_i = 0, i = 1, \dots, n \Leftrightarrow \begin{cases} s_B \geq 0, s_i x_i = 0, i \in B \\ s_N \geq 0, s_i x_i = 0, i \in N \end{cases} \begin{array}{l} \rightarrow \text{vérifié en prenant } s_B = 0 \\ \rightarrow \text{vérifié car } x_N = 0 \end{array}$$

$$\begin{cases} B^T \lambda = c_B \\ N^T \lambda + s_N = c_N \\ s_N \geq 0 \end{cases} \Rightarrow \begin{cases} \lambda = B^{-T} c_B \\ s_N = c_N - (B^{-1} N)^T c_B = \bar{c}_N \geq 0 \end{cases} \Rightarrow s = \begin{pmatrix} s_B \\ s_N \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{c}_N \end{pmatrix} = \bar{c} \geq 0$$

- Les coûts réduits sont les multiplicateurs des variables $\rightarrow s = \bar{c} \geq 0$

1.4.5 Problème quadratique

- ☐ Forme standard
- ☐ Conditions nécessaires d'optimalité
- ☐ Projection
- ☐ Directions conjuguées

1.4.5 Problème quadratique

Problème quadratique

- Forme standard $\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + c^T x$ sous $Ax = b$ $Q \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$
- Gradient : $g(x) = Qx + c$
- Hessien : $H(x) = Q$

Cas d'une matrice Q non symétrique

On se peut toujours se ramener à une **matrice Q' symétrique** : $Q'_{ij} = Q'_{ji} = \frac{1}{2}(Q_{ij} + Q_{ji})$

$$\Rightarrow x^T Q x = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (Q_{ij} + Q_{ji}) x_i x_j = \sum_{i=1}^n \sum_{j=1}^n Q'_{ij} x_i x_j = x^T Q' x$$

Cas d'une matrice Q symétrique

- Q admet **n valeurs propres réelles** (distinctes ou non)
- Q admet une **base orthonormée de vecteurs propres**
- Si Q est **définie positive**, elle admet une factorisation LDL^T (**factorisation de Cholesky**)

1.4.5 Problème quadratique

Conditions nécessaires d'optimalité

- Lagrangien : $L(x, \lambda) = \frac{1}{2} x^T Q x + c^T x + \lambda^T (b - Ax) \Rightarrow \begin{cases} \nabla_x L(x, \lambda) = Qx + c - A^T \lambda \\ \nabla_{xx}^2 L(x, \lambda) = Q \end{cases}$
- Conditions d'ordre 2 : Q définie positive $\rightarrow Q$ inversible
- Conditions d'ordre 1 : $\begin{cases} Qx - A^T \lambda = -c \\ Ax = b \end{cases} \Rightarrow \begin{cases} x - Q^{-1} A^T \lambda = -Q^{-1} c \\ Ax = b \end{cases} \Rightarrow \begin{cases} Ax - AQ^{-1} A^T \lambda = -AQ^{-1} c \\ Ax = b \end{cases}$

Solution

Par soustraction membre à membre : $AQ^{-1} A^T \lambda = AQ^{-1} c + b \Rightarrow \lambda = (AQ^{-1} A^T)^{-1} (AQ^{-1} c + b)$

En reportant dans l'expression de x :

$$\begin{cases} \lambda = (AQ^{-1} A^T)^{-1} (AQ^{-1} c + b) \\ x = Q^{-1} (A^T \lambda - c) = Q^{-1} A^T (AQ^{-1} A^T)^{-1} (AQ^{-1} c + b) - Q^{-1} c \end{cases}$$

Application

Projection d'un vecteur sur un hyperplan

1.4.5 Projection

Projection d'un vecteur sur un hyperplan

La projection de $x_0 \in \mathbb{R}^n$ sur l'hyperplan d'équation $Ax=b$ est le point x solution de

$$\min_{x \in \mathbb{R}^n} \|x - x_0\| \text{ sous } Ax = b$$

→ point x_P de l'hyperplan le plus proche de x_0

- Problème quadratique équivalent

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|x - x_0\|^2 \text{ sous } Ax = b$$

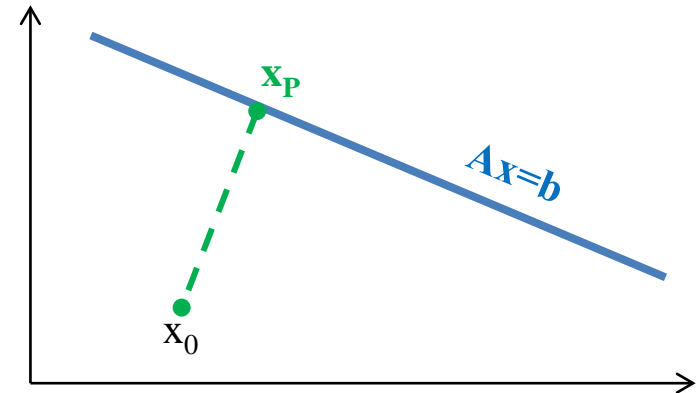
$$\Leftrightarrow \min_{x \in \mathbb{R}^n} \frac{1}{2} (x - x_0)^T (x - x_0) \text{ sous } Ax = b$$

- Lagrangien : $L(x, \lambda) = \frac{1}{2} (x - x_0)^T (x - x_0) + \lambda^T (b - Ax)$

- Condition d'ordre 1 : $\begin{cases} x - x_0 - A^T \lambda = 0 \\ Ax = b \end{cases} \Rightarrow \begin{cases} Ax - AA^T \lambda = Ax_0 \\ Ax = b \end{cases} \Rightarrow \begin{cases} \lambda = (AA^T)^{-1} (b - Ax_0) \\ x = x_0 + A^T (AA^T)^{-1} (b - Ax_0) \end{cases}$

Solution

$$x_P = \left(I - A^T (AA^T)^{-1} A \right) x_0 + A^T (AA^T)^{-1} b \rightarrow \text{matrice de projection : } \boxed{P = I - A^T (AA^T)^{-1} A}$$



1.4.5 Directions conjuguées

Forme quadratique définie positive

On considère une forme quadratique définie positive

$$f(x) = \frac{1}{2} x^T Q x + c^T x \quad \text{avec } Q \in \mathbb{R}^{n \times n} \text{ symétrique définie positive}$$

Relation de conjugaison

- 2 vecteurs u et v de \mathbb{R}^n sont **conjugués** par rapport à Q si : $u^T Q v = 0$
- n vecteurs $(d_i)_{i=1, \dots, n}$ conjugués 2 à 2 forment une **base** de \mathbb{R}^n

Preuve

On cherche $(\alpha_i)_{i=1, \dots, n}$ tels que : $\sum_{i=1}^n \alpha_i d_i = 0$

$$\Rightarrow d_k^T Q \sum_{i=1}^n \alpha_i d_i = 0, \forall k \Rightarrow \sum_{i=1}^n \alpha_i d_k^T Q d_i = 0, \forall k \Rightarrow \alpha_k d_k^T Q d_k = 0, \forall k \quad \text{car } d_k^T Q d_i \text{ si } i \neq k$$

Q définie positive : $d_k^T Q d_k > 0$ si $d_k \neq 0 \Rightarrow \alpha_k = 0$

Les n vecteurs $(u_i)_{i=1, \dots, n}$ sont indépendants \rightarrow base de \mathbb{R}^n

- Tout vecteur x de \mathbb{R}^n peut s'écrire : $x = \sum_{i=1}^n \alpha_i d_i$ avec $\alpha_i = \frac{d_i^T Q x}{d_i^T Q d_i}$

1.4.5 Directions conjuguées

Méthode de directions conjuguées

On obtient le minimum de la forme quadratique définie positive $f(x) = \frac{1}{2}x^T Qx + c^T x$ en **n itérations à pas optimal suivant des directions conjuguées**.

Preuve : On part du point initial x_0 : $x_0 = \sum_{i=1}^n \frac{d_i^T Qx_0}{d_i^T Qd_i} d_i$ dans la base $(d_i)_{i=1,\dots,n}$

*Après k itérations à pas optimal α_k suivant d_1, \dots, d_k , on obtient : $x_k = x_0 + \sum_{i=1}^k \alpha_i d_i = x_{k-1} + \alpha_k d_k$
 A l'itération k , le pas optimal α_k suivant la direction d_k vérifie :*

$$\min_{\alpha} f(x_{k-1} + \alpha d_k) \Rightarrow \frac{d}{d\alpha} f(x_{k-1} + \alpha d_k) = 0 \Rightarrow d_k^T \nabla f(x_{k-1} + \alpha d_k) = 0 \Rightarrow d_k^T \nabla f(x_k) = 0$$

$$\nabla f(x) = Qx + c \Rightarrow d_k^T \nabla f(x_k) = d_k^T Q \left(x_0 + \sum_{i=1}^k \alpha_i d_i \right) + d_k^T c = d_k^T (Qx_0 + c) + \alpha_k d_k^T Qd_k$$

$$d_k^T \nabla f(x_k) = 0 \Rightarrow \alpha_k = -\frac{d_k^T (Qx_0 + c)}{d_k^T Qd_k}$$

On obtient pour x_n :

$$\begin{aligned} x_n &= x_0 + \sum_{i=1}^n \alpha_i d_i = x_0 - \sum_{i=1}^n \frac{d_i^T (Qx_0 + c)}{d_i^T Qd_i} d_i = x_0 - \sum_{i=1}^n \frac{d_i^T Qx_0}{d_i^T Qd_i} d_i - \sum_{i=1}^n \frac{d_i^T Q(Q^{-1}c)}{d_i^T Qd_i} d_i \\ &= x_0 - x_0 - Q^{-1}c \quad \text{dans la base } (d_i) \end{aligned}$$

$$\Rightarrow x_n = -Q^{-1}c \Rightarrow \nabla f(x_n) = Q^{-1}x_n + c = 0 \rightarrow x_n = x^* \text{ minimum de } f$$

Sommaire

1. Bases théoriques
- 2. Optimisation sans contraintes**
 - 2.1 Méthodes de descente
 - 2.2 Méthode de Newton
 - 2.3 Recherche linéaire
 - 2.4 Région de confiance
 - 2.5 Moindres carrés
 - 2.6 Méthode de Nelder-Mead
3. Optimisation avec contraintes

2 Optimisation sans contraintes

Problème non linéaire sans contraintes

$\min_{x \in \mathbb{R}^n} f(x) \rightarrow$ problème noté (PO)

Méthodes globales

- Capacité à localiser plusieurs minima locaux (éventuellement le minimum global)
- Algorithmes non déterministes (déplacements aléatoires « organisés »)
- Métaheuristiques : algorithmes génétiques, recuit simulé, essaims, colonies de fourmis, recherche tabou,...
- Convergence généralement lente, peu précise

Méthodes locales

- Recherche d'un **minimum local** à partir d'un **point initial** fourni par l'utilisateur
- Méthodes d'ordre 0 : sans dérivées \rightarrow Nelder-Mead
- d'ordre 1 : avec dérivées premières \rightarrow plus forte pente
- d'ordre 2 : avec dérivées premières et secondes \rightarrow Newton
- Critères d'efficacité : **rapidité** de convergence (nombre d'appels de la fonction)
précision de convergence
robustesse à l'initialisation

Sommaire

1. Bases théoriques
2. **Optimisation sans contraintes**
 - 2.1 **Méthodes de descente**
 - 2.1.1 Principes
 - 2.1.2 Itérations
 - 2.1.3 Initialisation et arrêt
 - 2.2 Méthode de Newton
 - 2.3 Recherche linéaire
 - 2.4 Région de confiance
 - 2.5 Moindres carrés
 - 2.6 Méthode de Nelder-Mead
3. Optimisation avec contraintes

2.1.1 Méthodes de descente

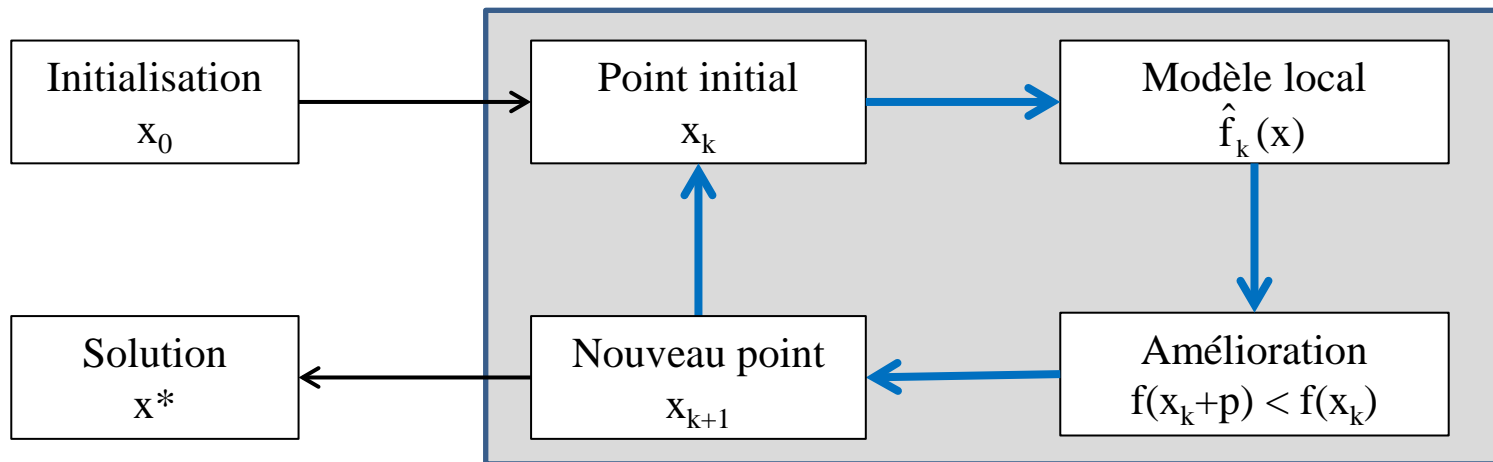
Problème sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x) \quad x^* \text{ minimum local} \Rightarrow \begin{cases} \nabla f(x^*) = 0 \\ \nabla^2 f(x^*) \geq 0 \end{cases}$$

On ne sait pas trouver le minimum global dans le cas général (f quelconque).

Méthode locale

- Initialisation x_0 → recherche d'un minimum local au voisinage de x_0
- Itérations → passage du point x_k au point x_{k+1} meilleur
- Arrêt → solution x^* ou blocage



2.1.2 Itérations

Modèle local : **prédiction**

- Point courant $x_k, f_k = f(x_k)$
- Evaluation de $g_k = \nabla f(x_k)$ ou approximation (différences finies)
 $H_k = \nabla^2 f(x_k)$ ou approximation (quasi Newton)
- Modèle quadratique : $\min_p \hat{f}_k(x_k + p) = f_k + p^t g_k + \frac{1}{2} p^t H_k p \rightarrow \hat{x}_{k+1} = x_k + \hat{p}$
(prédiction)

→ **Méthodes de Newton ou quasi-Newton**

Amélioration : **correction**

- Nouveau point $x_{k+1} = x_k + p$ tel que $f(x_k + p) < f(x_k)$
- Déplacement p à partir de x_k
par recherche linéaire suivant $d_k = \hat{x}_{k+1} - x_k$
par région de confiance dans $\|x - x_k\| < r$

→ **Méthodes de globalisation**

- La méthode de Newton appliquée directement ne converge pas systématiquement.
La globalisation est nécessaire pour contrôler la convergence.

2.1.3 Initialisation et arrêt

Initialisation

- Les **méthodes locales** recherchent le minimum au voisinage du point de départ.
- Objectifs :
 - rapidité de convergence
 - précision de convergence→ **méthodes à base de dérivées**
- Le minimum local trouvé est le plus proche du point initial x_0 .
 - initialisation à modifier pour trouver un autre minimum local
- Les **méthodes « globales »** explorent « aléatoirement » les solutions
 - localisation possible de plusieurs minima locaux

Conditions d'arrêt

- Déplacement insuffisant : $\|x_{k+1} - x_k\| < \varepsilon_x$
- Amélioration insuffisante : $f_k - f_{k+1} < \varepsilon_f$
- Condition d'ordre 1 vérifiée : $\|g_k\| < \varepsilon_g$
- Nombre maximal d'itérations ou d'appels fonction : $N_{\text{iter}}, N_{\text{fonc}}$

Sommaire

1. Bases théoriques
2. **Optimisation sans contraintes**
 - 2.1 Méthodes de descente
 - 2.2 **Méthode de Newton**
 - 2.2.1 Résolution d'équations
 - 2.2.2 Minimisation
 - 2.2.3 Globalisation
 - 2.3 Recherche linéaire
 - 2.4 Région de confiance
 - 2.5 Moindres carrés
 - 2.6 Méthode de Nelder-Mead
3. Optimisation avec contraintes

2.2.1 Résolution d'équations

- ☐ Principes
- ☐ Méthode de Newton
- ☐ Méthode de quasi-Newton

2.2.1 Résolution d'équations

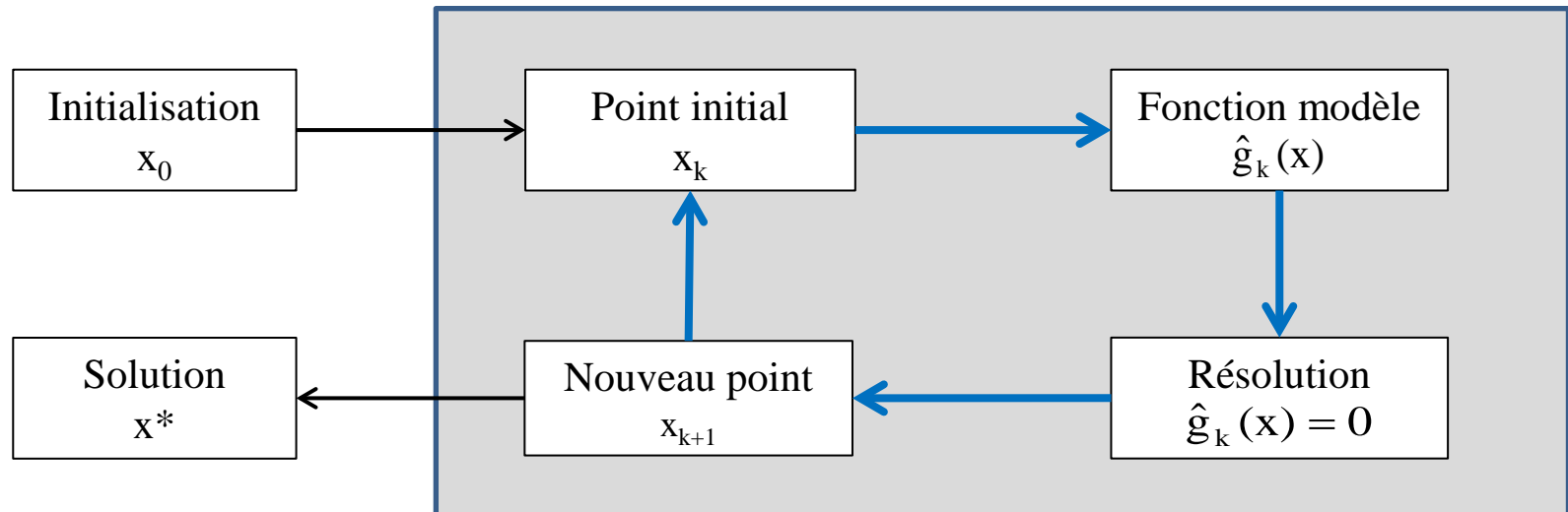
Système d'équations non linéaires

$$g(x) = 0 \quad \text{avec } g : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

→ système de n équations à n inconnues

Principe de la méthode de Newton

- Linéariser g au point initial x_0 → fonction modèle linéaire \hat{g} « proche » de g
- Résoudre le système linéaire $\hat{g}(x)=0$ → nouveau point x_1
- Itérer jusqu'à vérifier $g(x)=0$ → solution x^*



2.2.1 Méthode de Newton

Fonction modèle

- Développement de Taylor à l'ordre 1 de g en x_k

$$g(x) = g(x_k) + \nabla g(x_k)^T (x - x_k) + o(\|x - x_k\|)$$

- Modèle linéaire de g en x_k :

$$\hat{g}_k(x) = g(x_k) + G_k(x - x_k) \quad \text{avec } G_k \in \mathbb{R}^{n \times n}$$

Choix de la matrice G_k

- Méthode de **Newton** $\rightarrow G_k = \nabla g(x_k)^T =$ matrice jacobienne de g en x_k
- Méthode de **quasi-Newton** $\rightarrow G_k =$ approximation de $\nabla g(x_k)^T$

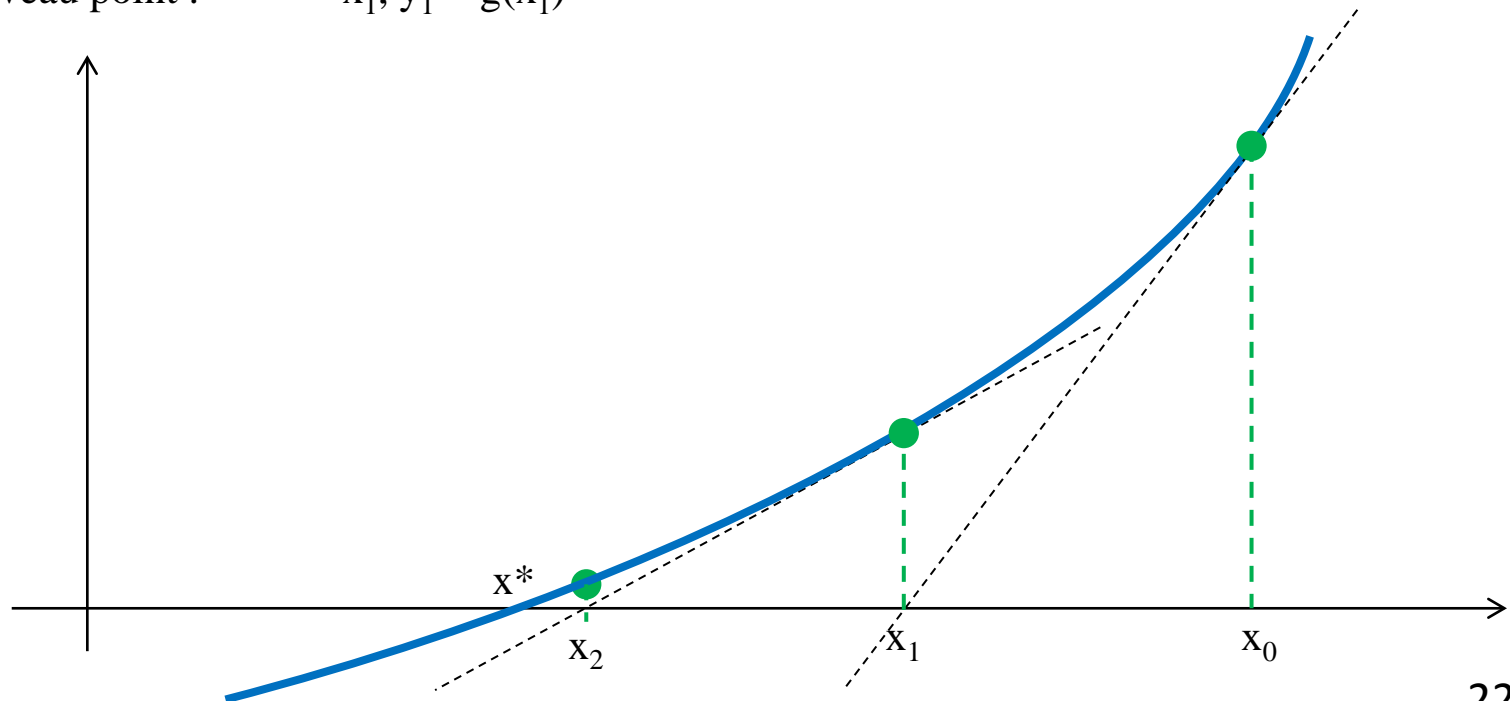
Résolution

- Système linéaire : $\hat{g}_k(x) = 0 \Rightarrow g(x_k) + G_k(x - x_k) = 0$
- Itération : $x_{k+1} = x_k - G_k^{-1}g(x_k)$ si G_k inversible
- Condition d'arrêt : $\|g(x_k)\| < \varepsilon$

2.2.1 Méthode de Newton

Illustration à une variable

- Equation non linéaire : $g(x) = 0$
- Point initial : $x_0, y_0 = g(x_0) \rightarrow$ **Tangente** en x_0 : $y = y_0 + g'(x_0)(x - x_0)$
- Intersection axe x : $y = 0 \Rightarrow x_1 = x_0 - \frac{y_0}{g'(x_0)}$
- Nouveau point : $x_1, y_1 = g(x_1)$



2.2.1 Méthode de Newton

Modèle linéaire de g en x_k

$$\hat{g}_k(x) = g(x_k) + G_k(x - x_k) \quad \text{avec } G_k = \nabla g(x_k)^T$$

Erreur de linéarisation

M = constante de Lipschitz sur le gradient \approx majorant de la courbure

$$\|g(x) - \hat{g}_k(x)\| \leq \frac{1}{2} M \|x - x_k\|^2 \rightarrow \text{erreur quadratique}$$

Vitesse de convergence

Hypothèses sur la solution x^* : $\nabla g(x^*)$ inversible

$$\|\nabla g(x^*)^{-1}\| \leq \eta$$

Suite (x_k) : $x_{k+1} = x_k - G_k^{-1} g(x_k)$

- (x_k) converge vers x^* si x_0 est « assez proche » de x^* : $\exists r > 0 / \|x_0 - x^*\| < r \Rightarrow \lim_{k \rightarrow \infty} x_k = x^*$

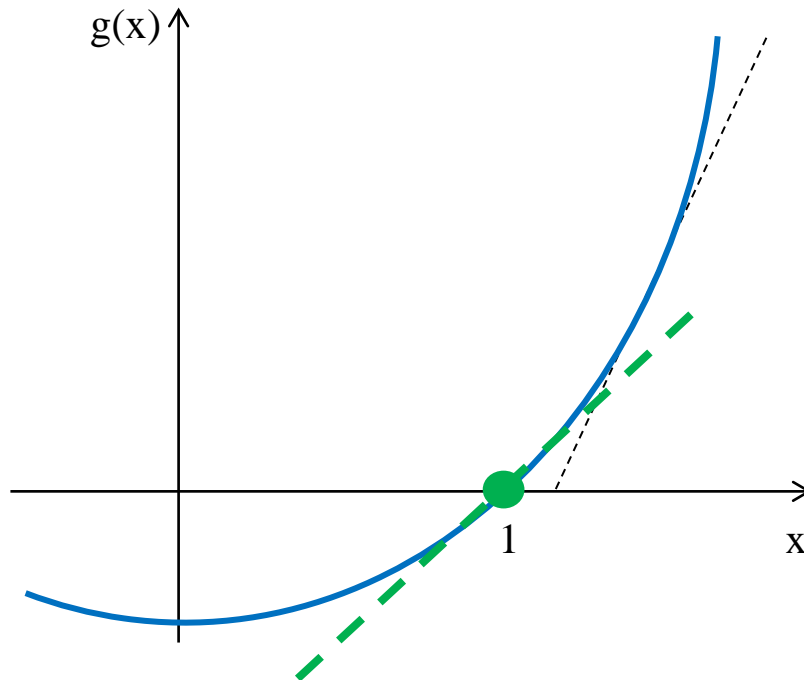
- La convergence est **quadratique**

$$\rightarrow \|x_{k+1} - x^*\| \leq M\eta \|x_k - x^*\|^2$$

2.2.1 Exemples

Exemple 1

- Fonction : $g(x) = x^2 - 1$
- Dérivée : $g'(x) = 2x$
- Solution : $x = 1 \rightarrow g'(1) = 2$



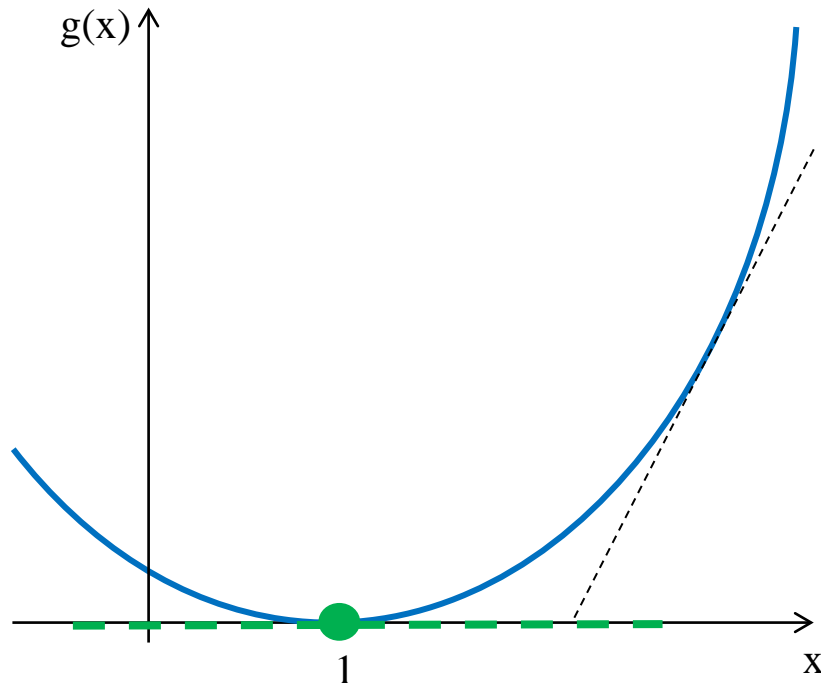
Iteration	x(k)	g(x)=x**2-1	g'(x)=2x	Erreur
0	4,00000000	1,5E+01	8,0000	3,0E+00
1	2,12500000	3,5E+00	4,2500	1,1E+00
2	1,29779412	6,8E-01	2,5956	3,0E-01
3	1,03416618	6,9E-02	2,0683	3,4E-02
4	1,00056438	1,1E-03	2,0011	5,6E-04
5	1,00000016	3,2E-07	2,0000	1,6E-07
6	1,00000000	2,5E-14	2,0000	1,3E-14

Convergence quadratique
 $g'(x^*)$ inversible

2.2.1 Exemples

Exemple 2

- Fonction : $g(x) = (x - 1)^2$
- Dérivée : $g'(x) = 2(x - 1)$
- Solution : $x = 1 \rightarrow \mathbf{g'(1) = 0}$



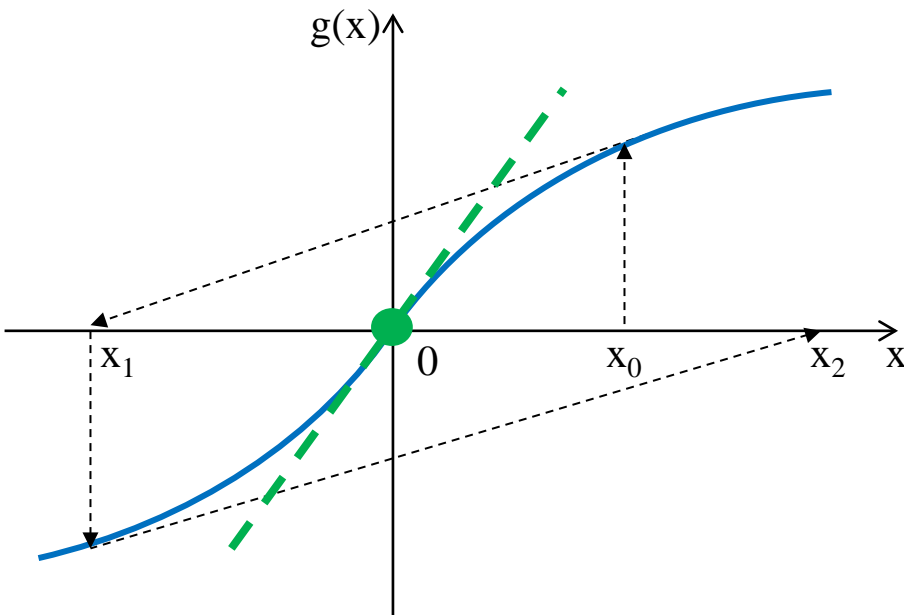
Iteration	x(k)	g(x)=(x-1)**2	g'(x)=2(x-1)	Erreur
0	4,00000000	9,0E+00	6,0000	3,0E+00
1	2,50000000	2,3E+00	3,0000	1,5E+00
2	1,75000000	5,6E-01	1,5000	7,5E-01
3	1,37500000	1,4E-01	0,7500	3,8E-01
4	1,18750000	3,5E-02	0,3750	1,9E-01
5	1,09375000	8,8E-03	0,1875	9,4E-02
6	1,04687500	2,2E-03	0,0938	4,7E-02
7	1,02343750	5,5E-04	0,0469	2,3E-02
8	1,01171875	1,4E-04	0,0234	1,2E-02
9	1,00585938	3,4E-05	0,0117	5,9E-03
10	1,00292969	8,6E-06	0,0059	2,9E-03
15	1,00009155	8,4E-09	0,0002	9,2E-05
20	1,00000286	8,2E-12	0,0000	2,9E-06

Convergence lente
 $g'(x^*)$ non inversible

2.2.1 Exemples

Exemple 3

- Fonction : $g(x) = \text{Arc tan}(x)$
- Dérivée : $g'(x) = \frac{1}{1+x^2}$
- Solution : $x = 0 \rightarrow g''(1) = 0$



Iteration	x(k)	g(x)=Arctan(x)	g'(x)=1/(1+x**2)	Erreur
0	1,300	0,915	0,372	1,3E+00
1	-1,162	-0,860	0,426	-1,2E+00
2	0,859	0,710	0,575	8,6E-01
3	-0,374	-0,358	0,877	-3,7E-01
4	0,034	0,034	0,999	3,4E-02
5	0,000	0,000	1,000	-2,6E-05
6	0,000	0,000	1,000	1,2E-14

Convergence

Iteration	x(k)	g(x)=Arctan(x)	g'(x)=1/(1+x**2)	Erreur
0	1,500	0,983	0,308	1,5E+00
1	-1,694	-1,038	0,258	-1,7E+00
2	2,321	1,164	0,157	2,3E+00
3	-5,114	-1,378	0,037	-5,1E+00
4	32,296	1,540	0,001	3,2E+01
5	-1575,317	-1,570	0,000	-1,6E+03
6	3804976,008	1,571	0,000	3,9E+06

Divergence

2.2.1 Méthode de Newton

Intérêt de la méthode de Newton

- Convergence quadratique au voisinage de la solution → **très rapide et précise**
- Méthode à privilégier dans les algorithmes d'optimisation

Difficultés

- Calcul explicite du gradient $\nabla g(x_k)$ à chaque itération → coûteux (n appels fonctions)
- Convergence non garantie → même près de la solution

Adaptations

- Méthodes de **quasi-Newton** → $G_k =$ **approximation** du gradient $\nabla g(x_k)$
construite à partir des itérations précédentes
sans calcul explicite du gradient
- Techniques de **globalisation** → **Contrôle** du point x_{k+1} (meilleur que x_k ?)
 $\|g(x_{k+1})\| < \|g(x_k)\|$

Si le point x_{k+1} n'est pas satisfaisant → Méthodes de recherche linéaire
ou région de confiance
pour minimiser $\|g(x)\|^2$

2.2.1 Méthode de quasi-Newton


Méthode de Broyden


On cherche à définir la matrice G_k à partir de la matrice G_{k-1} de l'itération précédente.
 Les matrices G_{k-1} et G_k doivent être « proches » au sens de la norme matricielle.

Variation de modèle

- Modèle linéaire de g en x_{k-1} : $\hat{g}_{k-1}(x) = g(x_{k-1}) + G_{k-1}(x - x_{k-1})$
- Modèle linéaire de g en x_k : $\hat{g}_k(x) = g(x_k) + G_k(x - x_k)$
- Différence entre les modèles en x_{k-1} et x_k

$$\begin{aligned}
 \hat{g}_k(x) &= g(x_k) + G_k(x - x_k) \\
 &= g(x_{k-1}) + G_k(x_k - x_{k-1}) + G_k(x - x_k) \\
 &= g(x_{k-1}) + G_k(x - x_{k-1}) \\
 &= \hat{g}_{k-1}(x) - G_{k-1}(x - x_{k-1}) + G_k(x - x_{k-1}) \\
 &= \hat{g}_{k-1}(x) + (G_k - G_{k-1})(x - x_{k-1})
 \end{aligned}$$

 car $g(x_k) = g(x_{k-1}) + G_k(x_k - x_{k-1})$
 par définition de G_k

 car $\hat{g}_{k-1}(x) = g(x_{k-1}) + G_{k-1}(x - x_{k-1})$
 par définition de \hat{g}_{k-1}

$$\Rightarrow \hat{g}_k(x) - \hat{g}_{k-1}(x) = (G_k - G_{k-1})(x - x_{k-1})$$

2.2.1 Méthode de quasi-Newton

Objectif

Conserver un modèle linéaire de g en x_k

$$\hat{g}_k(x) = g(x_k) + G_k(x - x_k) \quad \text{avec } G_k \approx \nabla g(x_k)^T$$

sans calculer explicitement G_k

Equation de la sécante

On choisit une matrice $G_k \in \mathbb{R}^{n \times n}$ vérifiant :

$$g(x_k) - g(x_{k-1}) = G_k(x_k - x_{k-1}) \Leftrightarrow y_{k-1} = G_k d_{k-1} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

→ **équation de la sécante** entre x_{k-1} et x_k

Choix de G

Il existe une infinité de matrices G vérifiant l'équation de la sécante :

n^2 inconnues (composantes de $G \in \mathbb{R}^{n \times n}$)

n équations

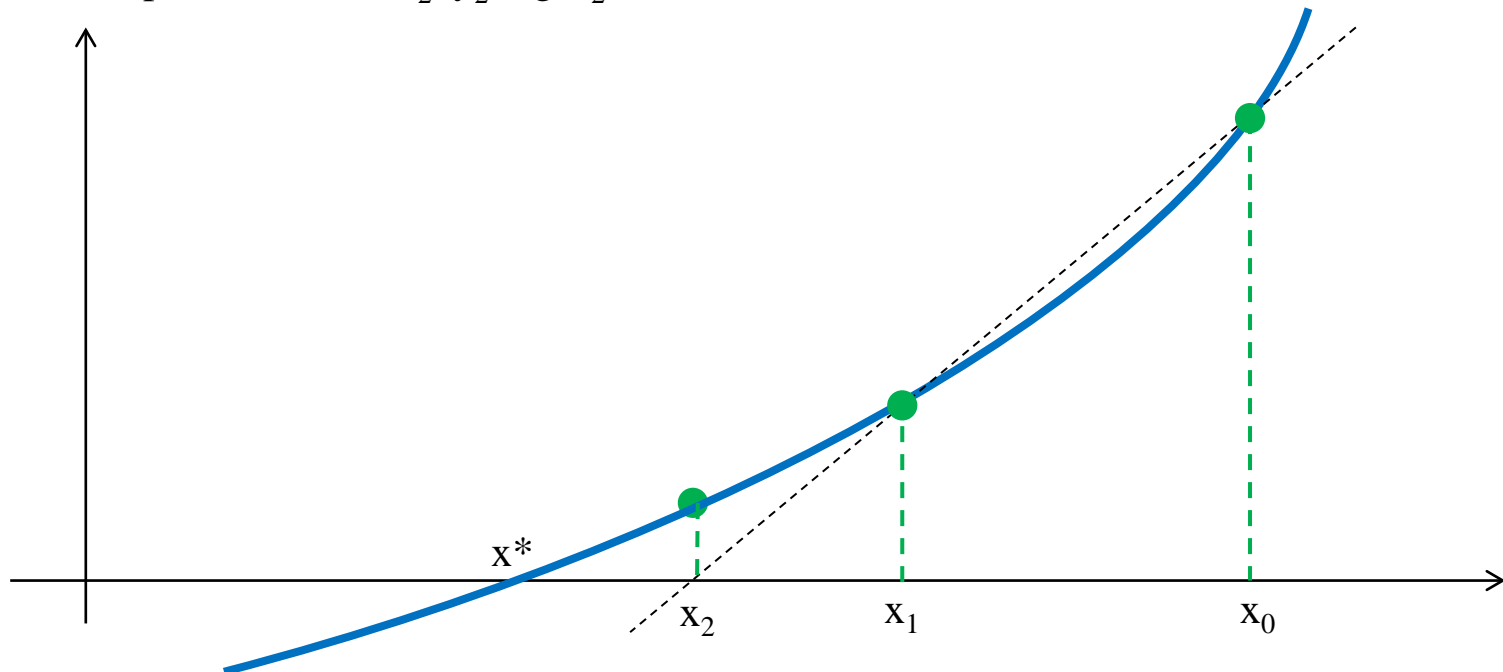
Chaque ligne de G définit un hyperplan de \mathbb{R}^n passant par x_{k-1} et x_k

→ infinité d'hyperplans possibles

2.2.1 Méthode de quasi-Newton

Illustration à une variable

- Equation non linéaire : $g(x) = 0$
- Points initiaux : $x_0, y_0 = g(x_0)$
 $x_1, y_1 = g(x_1)$ → **Sécante** en x_1 : $y = y_0 - \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$
- Intersection axe x : $y = 0 \Rightarrow x_2 = x_0 - \frac{x_1 - x_0}{y_1 - y_0} y_0$
- Nouveau point : $x_2, y_2 = g(x_2)$



2.2.1 Méthode de quasi-Newton

Mise à jour de Broyden

L'écart entre les modèles $\hat{g}_{k-1}(x)$ et $\hat{g}_k(x)$ est minimal en choisissant G_k solution de :

$$\min_{G \in \mathbb{R}^{n \times n}} \|G - G_{k-1}\| \text{ sous } y_{k-1} = G d_{k-1} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

Formule de Broyden :

$$G_k = G_{k-1} + \frac{(y_{k-1} - G_{k-1} d_{k-1}) d_{k-1}^T}{d_{k-1}^T d_{k-1}} \rightarrow \text{solution optimale}$$

Convergence

- La matrice G ne converge pas forcément vers $\nabla g \rightarrow$ ne compromet pas la convergence
- Les méthodes de quasi-Newton et de Newton peuvent converger vers des solutions différentes.
- La méthode de quasi-Newton converge généralement moins vite que la méthode de Newton, mais nécessite beaucoup moins d'appels de la fonction g (pas de calcul de gradient).
 - \rightarrow Peu de résultats théoriques
 - \rightarrow Méthode efficace en pratique, comportement à vérifier et adapter au cas par cas

2.2.1 Exemple

Comparaison Newton – Quasi-Newton

- Fonction : $g(x) = x^2 - 1$
- Dérivée : $g'(x) = 2x$
- Solution : $x = 1$

Quasi - Newton

Iteration	x(k)	$g(x)=x^{**2}-1$	dg/dx	Erreur
	5,000000000	2,4E+01		4,0E+00
0	4,000000000	1,5E+01	9,0000	3,0E+00
1	2,333333333	4,4E+00	6,3333	1,3E+00
2	1,63157895	1,7E+00	3,9649	6,3E-01
3	1,21238938	4,7E-01	2,8440	2,1E-01
4	1,04716672	9,7E-02	2,2596	4,7E-02
5	1,00443349	8,9E-03	2,0516	4,4E-03
6	1,00010193	2,0E-04	2,0045	1,0E-04
7	1,00000023	4,5E-07	2,0001	2,3E-07
8	1,000000000	2,3E-11	2,0000	1,1E-11

Newton

Iteration	x(k)	$g(x)=x^{**2}-1$	$g'(x)=2x$	Erreur
0	4,000000000	1,5E+01	8,0000	3,0E+00
1	2,12500000	3,5E+00	4,2500	1,1E+00
2	1,29779412	6,8E-01	2,5956	3,0E-01
3	1,03416618	6,9E-02	2,0683	3,4E-02
4	1,00056438	1,1E-03	2,0011	5,6E-04
5	1,00000016	3,2E-07	2,0000	1,6E-07
6	1,000000000	2,5E-14	2,0000	1,3E-14

2.2.2 Minimisation

- ☐ Principes
- ☐ Méthode de Newton
- ☐ Méthode de quasi-Newton
- ☐ Méthode BFGS
- ☐ Méthode DFP
- ☐ Méthode SR1

2.2.2 Problème de minimisation

Problème sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x) \quad \rightarrow \quad \begin{array}{l} \text{gradient : } g(x) = \nabla f(x) \\ \text{hessien : } H(x) = \nabla^2 f(x) \end{array}$$

Condition nécessaire de minimum local

$$x^* \text{ minimum local} \Rightarrow \begin{cases} g(x^*) = 0 \\ H(x^*) \geq 0 \end{cases} \quad (\text{hessien semi-défini positif})$$

Recherche des points stationnaires

Application de la méthode de Newton au système d'équations non linéaires : $\boxed{g(x) = 0}$

$$\text{Modèle linéaire de } g \text{ en } x_k : \hat{g}_k(x) = g(x_k) + G_k(x - x_k) \text{ avec } \begin{cases} g(x_k) = \nabla f(x_k) \\ G_k = \nabla g(x_k)^T = \nabla^2 f(x_k) = H_k \end{cases}$$

- Méthode de **Newton** : $G = H \rightarrow$ calcul explicite du hessien à chaque itération
- Méthode de **quasi-Newton** : $G =$ approximation de H
construite à partir des itérations précédentes
sans calcul explicite du hessien

2.2.2 Méthode de Newton

Modèle linéaire de $g=\nabla f$ en x_k

$$\hat{g}_k(x) = g(x_k) + G_k(x - x_k) \quad \text{avec} \quad \begin{cases} g(x_k) = \nabla f(x_k) \\ G_k = \nabla g(x_k)^T = \nabla^2 f(x_k) = H_k \end{cases}$$

- Itération : $x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \rightarrow$ équations de Newton
- Condition d'arrêt : $\|\nabla f(x_k)\| \leq \varepsilon$

Difficultés

- Calcul explicite et inversion du hessien $\nabla^2 f(x_k)$ à chaque itération \rightarrow coûteux
- Convergence non garantie même près de la solution
 \rightarrow mêmes difficultés que pour la résolution d'équations
- 1^{ère} condition nécessaire de minimum : $\nabla f(x^*)=0$
 \rightarrow point stationnaire $x^* =$ minimum local, maximum local ou point selle
 \rightarrow 2^{ème} condition nécessaire de minimum à vérifier : $\nabla^2 f(x^*) \geq 0$

Adaptations

- Méthodes de **quasi-Newton** $\rightarrow G_k =$ **approximation** du hessien $\nabla^2 f(x_k)$
- Techniques de **globalisation** \rightarrow **Contrôle** de la décroissance de f

2.2.2 Méthode de Newton

Modèle quadratique de f en x_k

- Développement de Taylor à l'ordre 2 de f en x_k

$$f(x) = f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2} (x - x_k)^T \nabla^2 f(x_k) (x - x_k) + o(\|x - x_k\|^2)$$

- Modèle quadratique en x_k

$$\hat{f}_k(x) = f_k(x_k) + g_k^T (x - x_k) + \frac{1}{2} (x - x_k)^T H_k (x - x_k)$$

- Lien entre le modèle de f et le modèle de $g = \nabla f$

$$\nabla \hat{f}_k(x) = g_k + H_k (x - x_k) = \hat{g}_k(x)$$

Minimisation du modèle de f en x_k

- Conditions suffisantes de minimum local : $\min_{x \in \mathbb{R}^n} \hat{f}_k(x) \Leftarrow \begin{cases} \nabla \hat{f}_k(x^*) = \hat{g}_k(x^*) = 0 \\ \nabla^2 \hat{f}_k(x^*) = H_k > 0 \end{cases}$

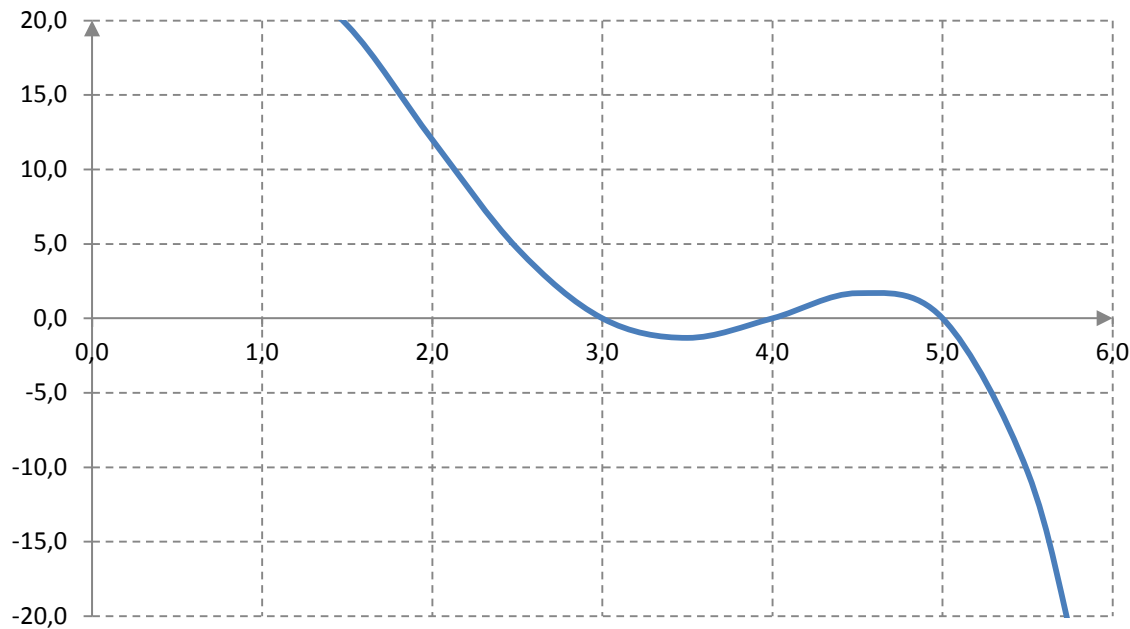
- Si le hessien de f en x_k est défini positif : $\nabla^2 f(x_k) > 0$
 Minimisation du modèle quadratique de f en x_k
 \Leftrightarrow Méthode de Newton en x_k pour résoudre $\nabla f(x) = 0$

- Sinon la méthode de Newton n'est pas directement applicable pour une minimisation

2.2.2 Exemple

Méthode de Newton

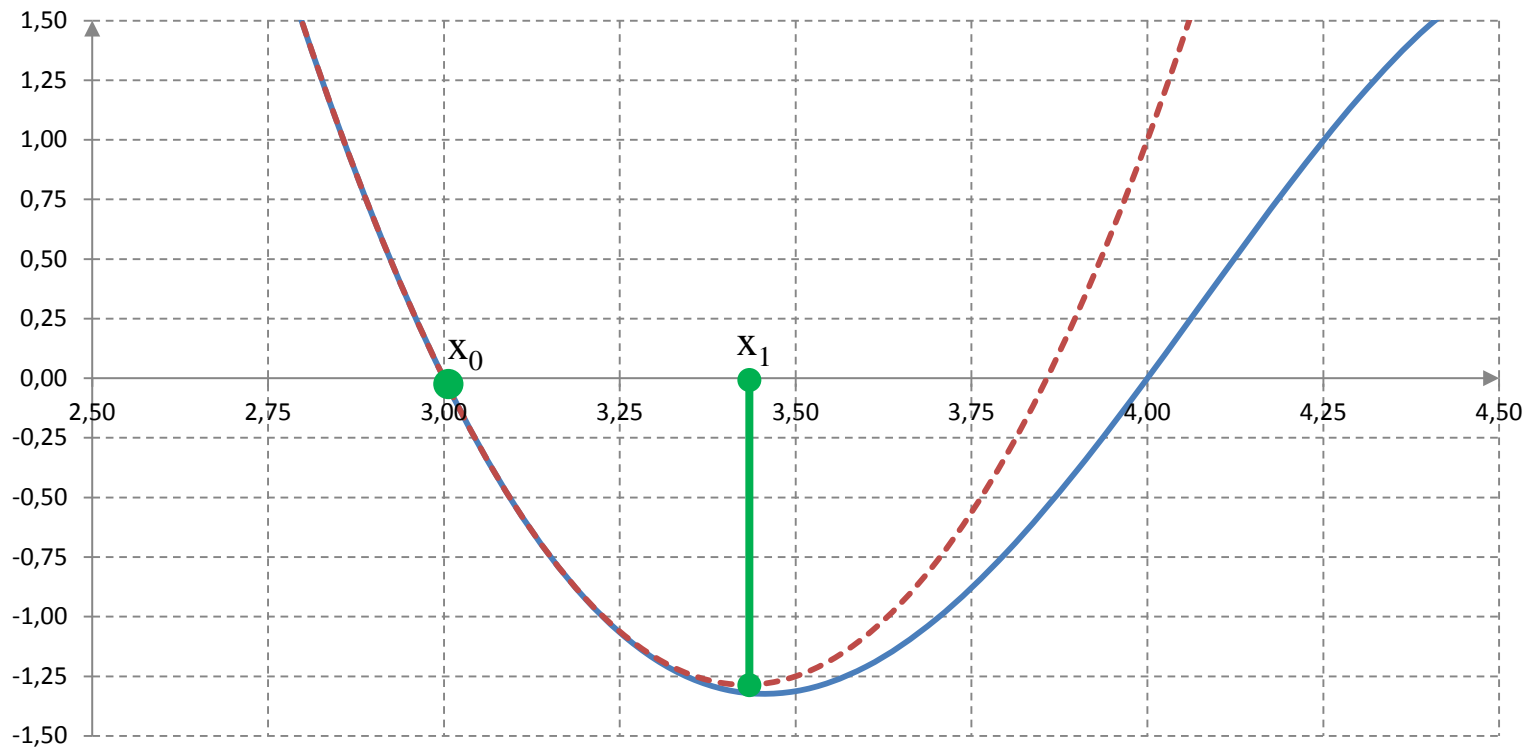
- Fonction : $f(x) = -x^4 + 12x^3 - 47x^2 + 60x$
- Dérivée : $f'(x) = -4x^3 + 36x^2 - 94x + 60 \rightarrow 3 \text{ zéros}$
 $\rightarrow 1 \text{ minimum local, } 2 \text{ maxima locaux}$



2.2.2 Exemple

Méthode de Newton

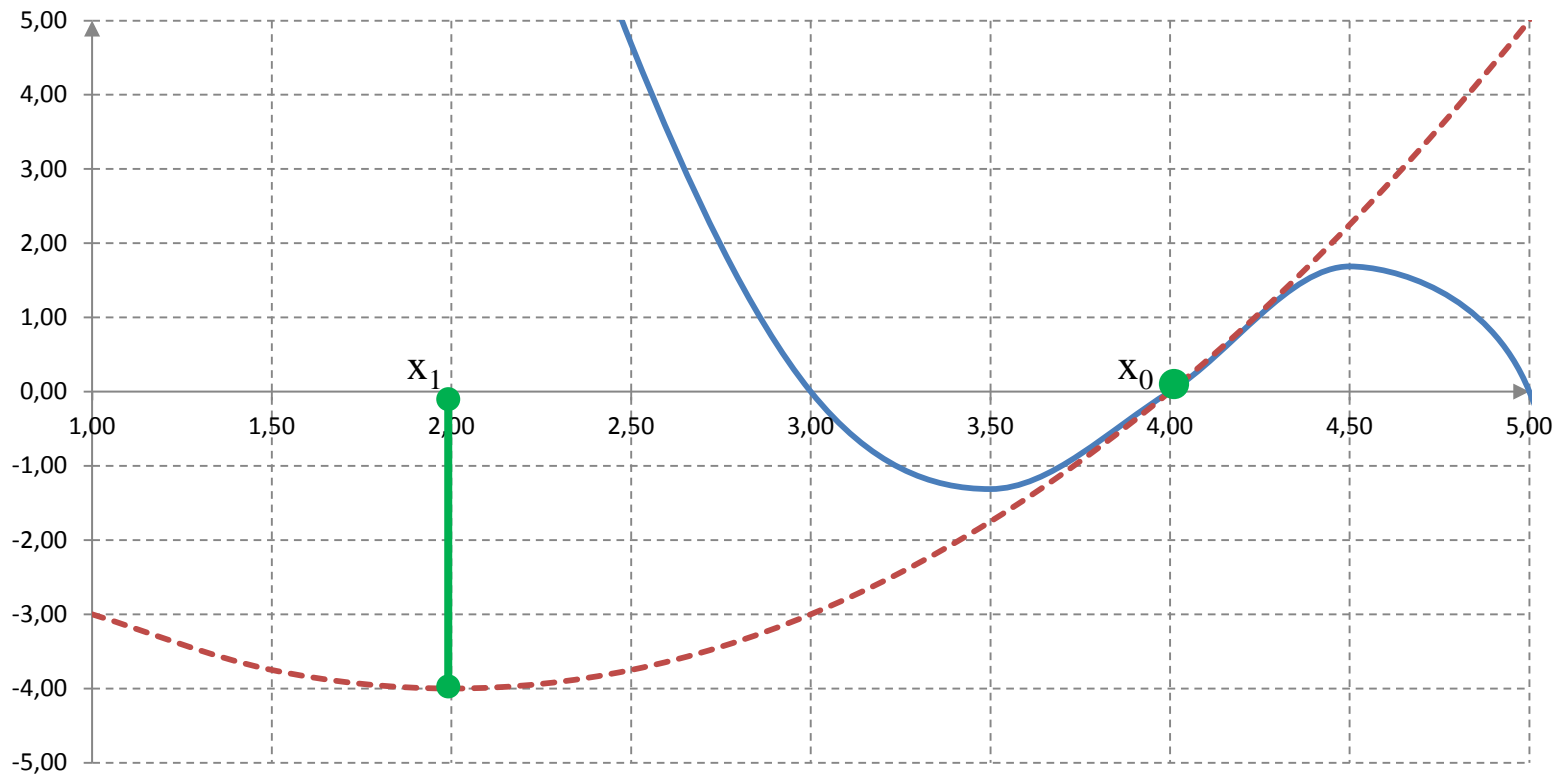
- Modèle quadratique en $x_0 = 3$: $\hat{f}_0(x) = 7x^2 - 48x + 81$
- Itération de Newton en $x_0 = 3$: $\min_x \hat{f}_0(x) \rightarrow x_1 = \frac{24}{7} \rightarrow$ Meilleur que x_0
 $f(x_1) = -1.32 < 0 = f(x_0)$



2.2.2 Exemple

Méthode de Newton

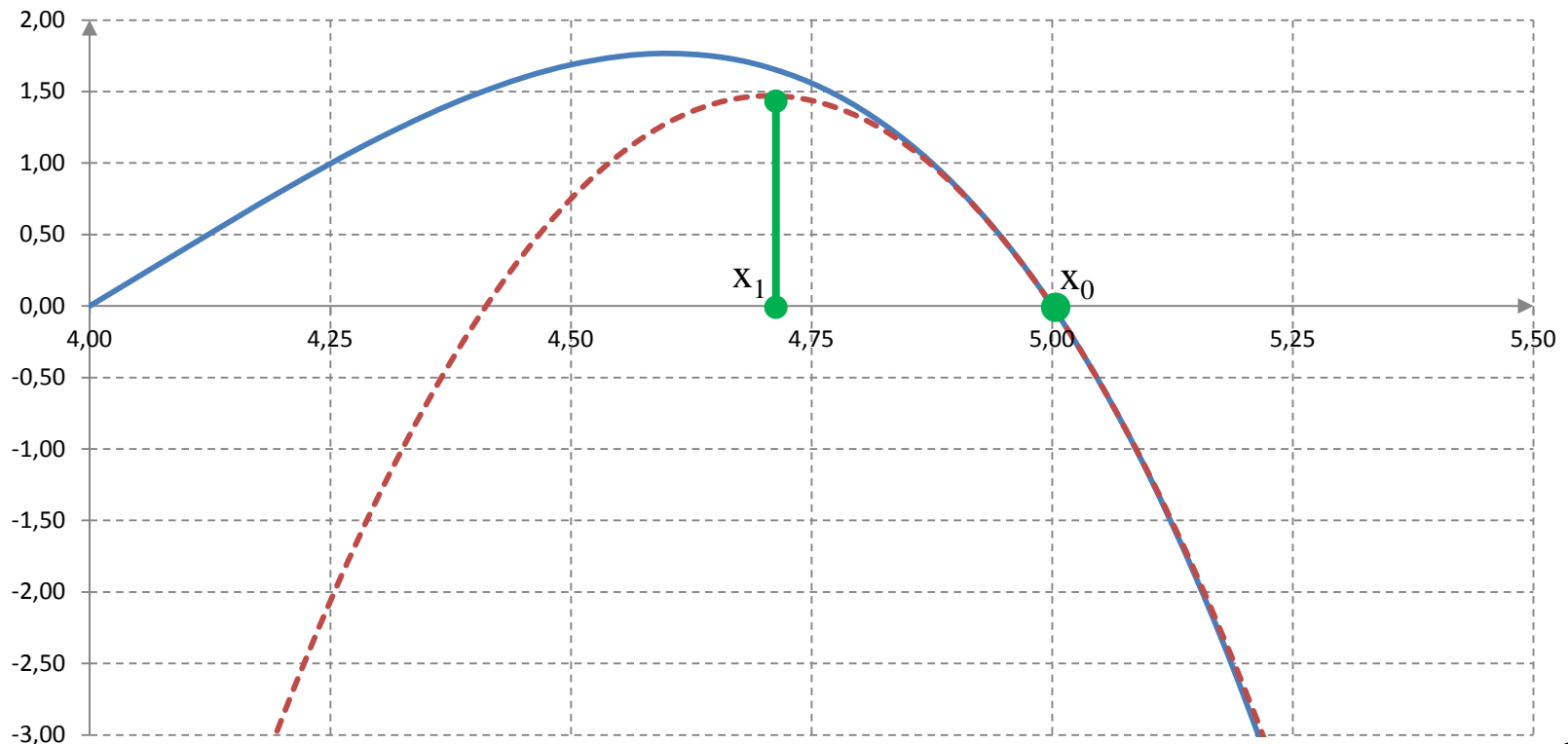
- Modèle quadratique en $\mathbf{x}_0 = 4$: $\hat{f}_0(x) = x^2 - 4x$
- Itération de Newton en $x_0 = 4$: $\min_x \hat{f}_0(x) \rightarrow x_1 = 2 \rightarrow$ Moins bon que x_0
 $f(x_1) = 12 > 0 = f(x_0)$



2.2.2 Exemple

Méthode de Newton

- Modèle quadratique en $x_0 = 5$: $\hat{f}_0(x) = -17x^2 + 160x - 375$
- Itération de Newton en $x_0 = 5$: $\min_x \hat{f}_0(x) \rightarrow x_1 = \frac{81}{17} \rightarrow$ Moins bon que x_0 (maximise f)
 $f(x_1) = 1.513 > 0 = f(x_0)$



2.2.2 Méthode de quasi-Newton

Objectif

Conserver un modèle quadratique de f en x_k

$$\hat{f}_k(x) = f_k(x_k) + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k (x - x_k) \quad \text{avec } H_k \approx \nabla^2 f(x_k)$$

sans calculer explicitement H_k

Mise à jour de Broyden

On peut appliquer la méthode de Broyden à la résolution de $g(x) = \nabla f(x) = 0$.

$$H_k = H_{k-1} + \frac{(y_{k-1} - H_{k-1}d_{k-1})d_{k-1}^T}{d_{k-1}^T d_{k-1}} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

Inconvénients

- H_k n'est pas forcément symétrique
- H_k n'est pas forcément positive

→ Modifications de la méthode si l'on souhaite avoir $H_k \approx \nabla^2 f(x_k)$

→ Formules BFGS, DFP ou SR1

2.2.2 Méthode BFGS

Equation sécante

Pour la **résolution d'équations**, on cherche la matrice H_k :

- vérifiant l'équation sécante $H_k d_{k-1} = y_{k-1}$
- la plus « proche » possible de H_{k-1} → **formule de Broyden**
- sans condition particulière sur la forme de la matrice

Pour une **minimisation**, on impose de plus à la matrice H_k d'être :

- symétrique
- définie positive → **formule BFGS**

Méthode de résolution

La matrice H_{k-1} issue de l'itération précédente est symétrique, définie positive.

- On part de la factorisation de Cholesky de H_{k-1} : $H_{k-1} = L_{k-1} L_{k-1}^T$
- On cherche la matrice H_k à partir de H_{k-1} sous la forme : $H_k = A_k A_k^T$

Il faut exprimer la matrice A_k en fonction de L_{k-1} .

→ On décompose l'équation sécante en 2 équations :

$$H_k d_{k-1} = y_{k-1} \Leftrightarrow A_k A_k^T d_{k-1} = y_{k-1} \Leftrightarrow \begin{cases} x = A_k^T d_{k-1} \\ A_k x = y_{k-1} \end{cases}$$

2.2.2 Méthode BFGS

Méthode de résolution de l'équation sécante

$$H_k d_{k-1} = y_{k-1} \Leftrightarrow A_k A_k^T d_{k-1} = y_{k-1} \Leftrightarrow \begin{cases} x = A_k^T d_{k-1} \\ A_k x = y_{k-1} \end{cases}$$

1. Pour x donné, on cherche A_k la plus « proche » possible de L_{k-1} vérifiant : $A_k x = y_{k-1}$
2. On détermine ensuite x en reportant l'expression de A_k dans : $x = A_k^T d_{k-1}$
3. On obtient $H_k = A_k A_k^T$ que l'on exprime en fonction de H_{k-1} .

Résolution

Notations sans indices : $L = L_{k-1}$, $A = A_k$
 $y = y_{k-1}$, $d = d_{k-1}$

1. En appliquant la formule de Broyden à L on obtient A en fonction de x : $A = L + \frac{(y - Lx)x^T}{x^T x}$
2. En reportant : $x = A^T d = L^T d + \frac{x(y - Lx)^T}{x^T x} d = L^T d + \frac{(y - Lx)^T d}{x^T x} x$

Il faut résoudre $x = L^T d + \frac{(y - Lx)^T d}{x^T x} x$ pour trouver x .

2.2.2 Méthode BFGS

Résolution de l'équation sécante

On cherche $x \in \mathbb{R}^n$ vérifiant : $x = L^T d + \frac{(y - Lx)^T d}{x^T x} x$

- Pour qu'une solution existe, le vecteur $L^T d$ doit être colinéaire à x :

$$x = \alpha L^T d \Rightarrow x^T x = \alpha^2 d^T H d$$

- En reportant dans l'équation : $\alpha L^T d = L^T d + \frac{(y - Lx)^T d}{\alpha^2 d^T H d} \alpha L^T d \Rightarrow \alpha^2 L^T d = \frac{y^T d}{d^T H d}$
- Pour qu'une solution existe, on doit avoir $y^T d > 0 \rightarrow A = L + \frac{1}{y^T d} \left(\alpha y d^T L - \frac{y^T d}{d^T H d} H d d^T L \right)$
 On obtient $H_k : H_k = A_k A_k^T$

Formule BFGS

- Mise à jour de H_k :
$$H_k = H_{k-1} + \frac{y_{k-1} y_{k-1}^T}{y_{k-1}^T d_{k-1}} - \frac{H_{k-1} d_{k-1} d_{k-1}^T H_{k-1}}{d_{k-1}^T H_{k-1} d_{k-1}} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

- Mise à jour **symétrique, de rang 2**
- Mise à jour **définie positive** si $y_{k-1}^T d_{k-1} > 0$

2.2.2 Méthode BFGS

Méthode de quasi Newton BFGS

- Le déplacement correspondant à une itération de la méthode de Newton est solution de

$$H_k d_k = -\nabla f(x_k) \Rightarrow d_k = -H_k^{-1} \nabla f(x_k)$$

→ La matrice utile pour l'itération de Newton est l'inverse de H_k .

- On inverse les 2 membres de la formule BFGS pour obtenir H_k^{-1} en fonction de H_{k-1}^{-1} .

$$H_k^{-1} = \left(I - \frac{d_{k-1} y_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) H_{k-1}^{-1} \left(I - \frac{y_{k-1} d_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) + \frac{d_{k-1} d_{k-1}^T}{d_{k-1}^T y_{k-1}} \quad \text{si } y_{k-1}^T d_{k-1} > 0 \quad \text{avec } \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

- Méthode élaborée par Broyden, Fletcher, Goldfarb, Shanno à la fin des années 1960
→ reconnue comme l'une des plus efficaces en pratique

Limitations

- Si la condition $y^T d > 0$ n'est pas vérifiée, on ne fait pas de mise à jour.
- Si le hessien n'est pas défini positif, la méthode BFGS ne converge pas vers le hessien
→ cas d'un hessien indéfini
→ optimisation avec contraintes (hessien réduit positif \neq hessien complet)

2.2.2 Méthode BFGS

Algorithme BFGS

- Direction de descente à l'itération k : $u_{k-1} = -H_{k-1}^{-1}g(x_{k-1})$
 - Minimisation dans la direction u_{k-1} : $x_k = x_{k-1} + su_{k-1}$ avec $\begin{cases} u_{k-1} = -H_{k-1}^{-1}g(x_{k-1}) \\ s \rightarrow \min_s f(x_{k-1} + su_{k-1}) \end{cases}$
 - Mise à jour de l'inverse du hessien
- $$H_k^{-1} = \left(I - \frac{d_{k-1}y_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) H_{k-1}^{-1} \left(I - \frac{y_{k-1}d_{k-1}^T}{d_{k-1}^T y_{k-1}} \right) + \frac{d_{k-1}d_{k-1}^T}{d_{k-1}^T y_{k-1}} \quad \text{si } y_{k-1}^T d_{k-1} > 0 \quad \text{avec } \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

Propriété 1

La mise à jour BFGS donne une matrice définie positive si $d_{k-1}^T y_{k-1} > 0$

Cette condition est réalisée si x_k est obtenu par **minimisation exacte** dans la direction $-H_{k-1}^{-1}g(x_{k-1})$

Propriété 2

Pour une fonction **f quadratique** : $f(x) = \frac{1}{2}x^T Qx + c^T x$

l'algorithme BFGS donne des directions successives vérifiant : $\begin{cases} u_i^T Q^{-1} u_j = 0 & , 0 \leq i \neq j \leq k \\ H_k^{-1} Q^{-1} u_i = u_i & , 0 \leq i \leq k \end{cases}$

→ **directions conjuguées** par rapport à Q^{-1}

→ **convergence en n itérations** avec à l'itération n : $H_n = Q$

2.2.2 Méthode DFP

Méthode de résolution

- Résolution de l'équation sécante sous la forme « inverse » : $H_k d_{k-1} = y_{k-1} \Leftrightarrow H_k^{-1} y_{k-1} = d_{k-1}$
- Mêmes principes que BFGS **appliqués à l'équation sécante inverse** pour obtenir H_k^{-1}
- Première méthode quasi-Newton élaborée par Davidon dans les années 1950

Formule DFP

- Mise à jour de l'inverse : $H_k^{-1} = H_{k-1}^{-1} + \frac{d_{k-1} d_{k-1}^T}{d_{k-1}^T y_{k-1}} - \frac{H_{k-1}^{-1} y_{k-1} y_{k-1}^T H_{k-1}^{-1}}{y_{k-1}^T H_{k-1}^{-1} y_{k-1}}$ avec $\begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$
- Mise à jour du hessien : $H_k = \left(I - \frac{y_{k-1} d_{k-1}^T}{y_{k-1}^T d_{k-1}} \right) H_{k-1} \left(I - \frac{d_{k-1} y_{k-1}^T}{y_{k-1}^T d_{k-1}} \right) + \frac{y_{k-1} y_{k-1}^T}{y_{k-1}^T d_{k-1}}$

Comparaison DFP – BFGS

- Formules identiques en permutant d_{k-1} et y_{k-1} pour mettre à jour le hessien ou l'inverse
- Mise à jour **symétrique, de rang 2**
- Mise à jour **définie positive** si $d_{k-1}^T y_{k-1} > 0 \rightarrow$ condition similaire à BFGS
- Méthode DFP en général moins efficace que BFGS

2.2.2 Méthode DFP

Algorithme DFP

- Direction de descente à l'itération k : $u_{k-1} = -H_{k-1}^{-1}g(x_{k-1})$
- Minimisation dans la direction u_{k-1} : $x_k = x_{k-1} + su_{k-1}$ avec $\begin{cases} u_{k-1} = -H_{k-1}^{-1}g(x_{k-1}) \\ s \rightarrow \min_s f(x_{k-1} + su_{k-1}) \end{cases}$
- Mise à jour de l'inverse du hessien

$$H_k^{-1} = H_{k-1}^{-1} + \frac{d_{k-1}d_{k-1}^T}{d_{k-1}^T y_{k-1}} - \frac{H_{k-1}^{-1}y_{k-1}y_{k-1}^T H_{k-1}^{-1}}{y_{k-1}^T H_{k-1}^{-1}y_{k-1}} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$

Propriété 1

La mise à jour DFP donne une matrice définie positive si $d_{k-1}^T y_{k-1} > 0$

Cette condition est réalisée si x_k est obtenu par **minimisation exacte** dans la direction $-H_{k-1}^{-1}g(x_{k-1})$

Propriété 2

Pour une fonction **f quadratique** : $f(x) = \frac{1}{2}x^T Qx + c^T x$

l'algorithme DFP donne des directions successives vérifiant :

$$\begin{cases} u_i^T Q u_j = 0 & , 0 \leq i \neq j \leq k \\ H_k^{-1} Q u_i = u_i & , 0 \leq i \leq k \end{cases}$$

→ **directions conjuguées** par rapport à Q

→ **convergence en n itérations** avec à l'itération n : $H_n = Q$

2.2.2 Méthode SR1

Equation sécante

La méthode SR1 construit une solution H_k de l'équation sécante $H_k d_{k-1} = y_{k-1}$

- **Symétrique, de rang 1** (i.e. dépendante d'un seul vecteur u de \mathbb{R}^n)
- Non nécessairement définie positive.

Méthode de résolution

- On cherche la matrice H_k à partir de H_{k-1} sous la forme

$$\boxed{H_k = H_{k-1} + uu^T} \rightarrow \text{addition d'une matrice symétrique de rang 1}$$

- Equation sécante : $y_{k-1} = H_k d_{k-1} = H_{k-1} d_{k-1} + uu^T d_{k-1} \Rightarrow y_{k-1} - H_{k-1} d_{k-1} = u^T d_{k-1} u$
- On pose : $\frac{1}{\gamma} = u^T d_{k-1} \Rightarrow y_{k-1} - H_{k-1} d_{k-1} = \frac{1}{\gamma} u \Rightarrow u = \gamma(y_{k-1} - H_{k-1} d_{k-1})$
- On reporte u pour obtenir γ : $\frac{1}{\gamma} = \gamma(y_{k-1} - H_{k-1} d_{k-1})^T d_{k-1} \Rightarrow d_{k-1}^T (y_{k-1} - H_{k-1} d_{k-1}) = \frac{1}{\gamma^2}$
- On exprime uu^T en fonction de $d_{k-1}, y_{k-1}, H_{k-1}$

$$\begin{cases} u = \gamma(y_{k-1} - H_{k-1} d_{k-1}) \\ \frac{1}{\gamma^2} = d_{k-1}^T (y_{k-1} - H_{k-1} d_{k-1}) \end{cases} \Rightarrow \begin{cases} uu^T = \gamma^2 (y_{k-1} - H_{k-1} d_{k-1})(y_{k-1} - H_{k-1} d_{k-1})^T \\ uu^T = \frac{(y_{k-1} - H_{k-1} d_{k-1})(y_{k-1} - H_{k-1} d_{k-1})^T}{d_{k-1}^T (y_{k-1} - H_{k-1} d_{k-1})} \end{cases}$$

2.2.2 Méthode SR1

Formule SR1

- Mise à jour de H_k :
$$H_k = H_{k-1} + \frac{(y_{k-1} - H_{k-1}d_{k-1})(y_{k-1} - H_{k-1}d_{k-1})^T}{d_{k-1}^T(y_{k-1} - H_{k-1}d_{k-1})} \quad \text{avec} \quad \begin{cases} d_{k-1} = x_k - x_{k-1} \\ y_{k-1} = g(x_k) - g(x_{k-1}) \end{cases}$$
- Mise à jour **symétrique, de rang 1**
- Mise à jour non nécessairement définie positive
→ Méthode alternative à la méthode BFGS (**cas d'un hessien indéfini**)

Limitations

- La formule SR1 peut donner des matrices H_k non définies positives, même si le hessien de la fonction est défini positif.
- Le dénominateur peut devenir petit → empêche la mise à jour et la convergence

Propriété

Pour une fonction **f quadratique** : $f(x) = \frac{1}{2}x^T Qx + c^T x$

la formule SR1 donne après n déplacements suivant des directions indépendantes d_k : $H_n = Q$
→ **ne nécessite pas de minimisation suivant d_k**

2.2.2 Comparaison

Méthodes de quasi-Newton BFGS – DFP – SR1

	BFGS	DFP	SR1
Matrice mise à jour	Hessien H_k	Inverse hessien H_k^{-1}	Hessien H_k
Equation résolue	Sécante	Inverse sécante	Sécante
Méthode	Broyden	Broyden	Résolution directe
Forme solution	Symétrique AA^T Rang 2 Définie positive	Symétrique AA^T Rang 2 Définie positive	Symétrique uu^T Rang 1 Indéfinie
Minimisation d_k	Précision moyenne	Précision forte	Précision faible
Fonction quadratique	Hessien exact : $H_n=Q$ Directions conjuguées Q^{-1}	Hessien exact : $H_n=Q$ Directions conjuguées Q	Hessien exact : $H_n=Q$
Limitations	Hessien de f indéfini	Hessien de f indéfini Précision minimisation	Matrices H_k non définies positives



Méthode BFGS réputée la plus efficace

2.2.2 Exemple

Méthode de quasi-Newton à une variable

Les formules BFGS et SR1 se simplifient pour une fonction à une variable.

- Fonction $f(x)$, $x \in \mathbb{R}$

- Mise à jour BFGS :
$$H_k = H_{k-1} + \frac{y_{k-1} y_{k-1}^T}{y_{k-1}^T d_{k-1}} - \frac{H_{k-1} d_{k-1} d_{k-1}^T H_{k-1}}{d_{k-1}^T H_{k-1} d_{k-1}}$$
$$= H_{k-1} + \frac{y_{k-1}}{d_{k-1}} - H_{k-1}$$
$$\Rightarrow H_k = \frac{y_{k-1}}{d_{k-1}} = \frac{g(x_k) - g(x_{k-1})}{x_k - x_{k-1}}$$

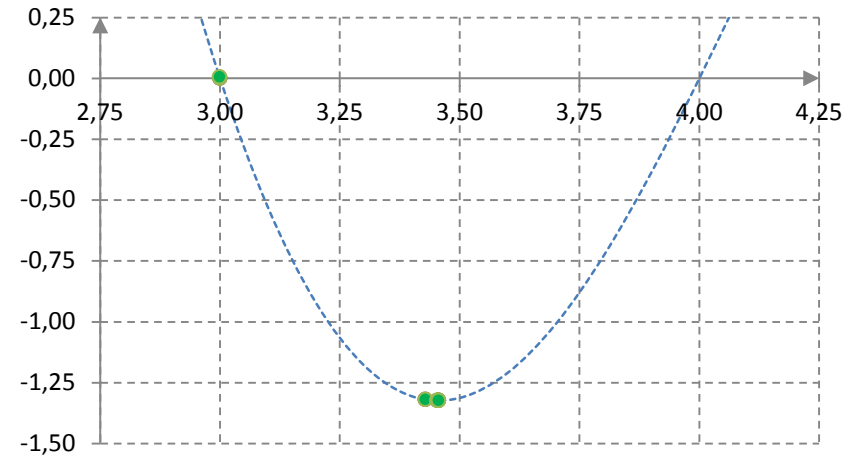
- Mise à jour SR1 :
$$H_k = H_{k-1} + \frac{(y_{k-1} - H_{k-1} d_{k-1})(y_{k-1} - H_{k-1} d_{k-1})^T}{d_{k-1}^T (y_{k-1} - H_{k-1} d_{k-1})}$$
$$= H_{k-1} + \frac{y_{k-1} - H_{k-1} d_{k-1}}{d_{k-1}}$$
$$\Rightarrow H_k = \frac{y_{k-1}}{d_{k-1}} = \frac{g(x_k) - g(x_{k-1})}{x_k - x_{k-1}}$$

→ On retrouve la formule de la sécante appliquée au gradient de f .

2.2.2 Exemple

Comparaison Newton – Quasi-Newton

- Fonction : $f(x) = -x^4 + 12x^3 - 47x^2 + 60x$
- Dérivée : $f'(x) = -4x^3 + 36x^2 - 94x + 60$
- Quasi-Newton : $h_k = \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$
- Point initial : $x_0 = 3 \rightarrow$ convergence
 Autres points \rightarrow divergence
 ou maximum de f



Quasi - Newton

Itération	x(k)	f(x)	f'(x)	h(k)	Erreur
0	3,00000000	0,00000000	-6,00E+00	1,000	-4,56E-01
1	2,99900000	0,00600700	-6,01E+00	14,000	-4,57E-01
2	3,42857155	-1,31945027	-3,15E-01	13,267	-2,70E-02
3	3,45230465	-1,32362420	-3,79E-02	11,672	-3,28E-03
4	3,45554876	-1,32368634	-4,68E-04	11,527	-4,06E-05
5	3,45558934	-1,32368635	-7,27E-07	11,509	-6,32E-08
6	3,45558940	-1,32368635	-1,40E-11	11,509	-1,22E-12
7	3,45558940	-1,32368635	-5,68E-14	11,462	-4,44E-15

Newton

Itération	x	f(x)	f'(x)	f''(x)	Erreur
0	3,00000000	0,00000000	-6,00E+00	14,000	-4,56E-01
1	3,42857143	-1,31945023	-3,15E-01	11,796	-2,70E-02
2	3,45526446	-1,32368574	-3,74E-03	11,513	-3,25E-04
3	3,45558935	-1,32368635	-5,77E-07	11,509	-5,01E-08
4	3,45558940	-1,32368635	-5,68E-14	11,509	-4,88E-15

2.2.2 Exemple

Méthode DFP à 2 variables

- Minimisation de $f(x) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \Rightarrow g(x) = \begin{pmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{pmatrix}$

- Point initial : $x_0 = (0 \ 0)$

- Itération 1** : $x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad H_0^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad g_0 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \rightarrow u_1 = -H_0^{-1}g_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

$$x_1 = x_0 + su_1 = \begin{pmatrix} -s \\ s \end{pmatrix} \rightarrow \min_s F(s) = s^2 - 2s \rightarrow s = 1 \rightarrow x_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \rightarrow g_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

- Mise à jour DFP de H^{-1}

$$\begin{cases} d_0 = x_1 - x_0 \\ y_0 = g(x_1) - g(x_0) \end{cases} \rightarrow d_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, y_0 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

$$H_1^{-1} = H_0^{-1} + \frac{d_0 d_0^T}{d_0^T y_0} - \frac{H_0^{-1} y_0 y_0^T H_0^{-1}}{y_0^T H_0^{-1} y_0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} - \frac{1}{4} \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix} = \boxed{\frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix}}$$

- Comparaison au vrai hessien : $H(x) = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix} \Rightarrow H^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$

2.2.2 Exemple

Méthode DFP à 2 variables

- **Itération 2** : $\mathbf{x}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ $\mathbf{H}_1^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix}$ $\mathbf{g}_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \rightarrow \mathbf{u}_2 = -\mathbf{H}_1^{-1} \mathbf{g}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$$\mathbf{x}_2 = \mathbf{x}_1 + s \mathbf{u}_2 = \begin{pmatrix} -1 \\ 1+s \end{pmatrix} \rightarrow \min_s F(s) = 1 - 3(1+s) + (1+s)^2 \rightarrow s = \frac{1}{2} \rightarrow \mathbf{x}_2 = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix} \rightarrow \mathbf{g}_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- On obtient le minimum en 2 itérations (fonction quadratique) : $\mathbf{x}^* = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$

- Mise à jour DFP de \mathbf{H}^{-1}

$$\begin{cases} \mathbf{d}_1 = \mathbf{x}_2 - \mathbf{x}_1 \\ \mathbf{y}_1 = \mathbf{g}(\mathbf{x}_2) - \mathbf{g}(\mathbf{x}_1) \end{cases} \rightarrow \mathbf{d}_1 = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}, \mathbf{y}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{H}_2^{-1} = \mathbf{H}_1^{-1} + \frac{\mathbf{d}_1 \mathbf{d}_1^T}{\mathbf{d}_1^T \mathbf{y}_1} - \frac{\mathbf{H}_1^{-1} \mathbf{y}_1 \mathbf{y}_1^T \mathbf{H}_1^{-1}}{\mathbf{y}_1^T \mathbf{H}_1^{-1} \mathbf{y}_1} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$$

- Comparaison au vrai hessien : $\mathbf{H}(\mathbf{x}) = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix} \Rightarrow \mathbf{H}^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$

2.2.2 Exemple

Méthode DFP à 2 variables

On vérifie les propriétés de la méthode DFP appliquée à une fonction quadratique.

$$f(\mathbf{x}) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 = \frac{1}{2} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \mathbf{Q} = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}$$

- Le minimum est obtenu en **2 itérations**.
- Les directions successives \mathbf{u}_1 et \mathbf{u}_2 sont **conjuguées** par rapport à \mathbf{Q}

$$\mathbf{u}_2^T \mathbf{Q} \mathbf{u}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} -2 \\ 0 \end{pmatrix} = 0$$

- On vérifie également :

$$\mathbf{H}_1^{-1} \mathbf{Q} \mathbf{u}_1 = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \mathbf{u}_1$$

$$\mathbf{H}_2^{-1} \mathbf{Q} \mathbf{u}_2 = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \mathbf{u}_2$$

avec $\boxed{\mathbf{H}_2^{-1} = \mathbf{Q}^{-1}}$

2.2.3 Globalisation

- ☐ Difficultés de la méthode de Newton
- ☐ Méthodes de globalisation
- ☐ Point de Newton et de Cauchy

2.2.3 Globalisation

Difficultés de la méthode de Newton

La convergence n'est pas garantie même près de la solution.

On ne peut appliquer directement l'itération de Newton.

→ techniques de globalisation pour vérifier et améliorer le point de Newton

Vérification du point de Newton

Le point x_{k+1} doit être meilleur que x_k pour être accepté.

- Pour une résolution d'équation : $g(x) = 0 \Rightarrow \|g(x_{k+1})\| < \|g(x_k)\|$
- Pour une minimisation : $\min_x f(x) = 0 \Rightarrow f(x_{k+1}) < f(x_k)$

Techniques de globalisation

Si le point x_N obtenu par l'itération de Newton ne vérifie pas les conditions d'amélioration, on procède à une recherche locale au voisinage de x_k .

- Méthode de **recherche linéaire** : suivant la direction du point de Newton x_N
- Méthode de **région de confiance** : à l'intérieur d'une sphère de centre x_k

2.2.3 Point de Newton et de Cauchy

Points particuliers

Deux points particuliers sont définis à partir du modèle quadratique de f en x_k :

$$\hat{f}_k(x) = f_k(x_k) + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k) \quad \text{avec} \quad \begin{cases} g_k = \nabla f_k(x_k) \\ H_k = \nabla^2 f_k(x_k) \end{cases}$$

- Point de Newton
 - Point de Cauchy
- points utiles dans les algorithmes de globalisation

Point de Newton

Le point de Newton x_N de f en x_k **minimise le modèle quadratique** en x_k .

x_N n'existe que si $\nabla^2 f(x_k)$ est définie positive.

$$x_N = x_k + d_N \quad \text{avec} \quad \boxed{\nabla^2 f(x_k) d_N = -\nabla f(x_k)} \quad \rightarrow d_N \text{ solution des équations de Newton}$$

Point de Cauchy

Le point de Cauchy x_C de f en x_k **minimise le modèle quadratique** en x_k **dans la direction $-g_k$** .

$$x_C = x_k - \alpha_C g_k \quad \text{solution de} \quad \min_{\alpha \geq 0} f(x_k - \alpha g_k) \quad \rightarrow \text{minimum suivant la plus forte descente}$$

Si f est convexe suivant $-g_k$:

$$\boxed{\alpha_C = \frac{g_k^T g_k}{g_k^T H_k g_k}}$$

Sommaire

1. Bases théoriques
2. **Optimisation sans contraintes**
 - 2.1 Méthodes de descente
 - 2.2 Méthode de Newton
 - 2.3 **Recherche linéaire**
 - 2.3.1 Principes
 - 2.3.2 Direction de descente
 - 2.3.3 Pas de déplacement
 - 2.3.4 Algorithme
 - 2.4 Région de confiance
 - 2.5 Moindres carrés
 - 2.6 Méthode de Nelder-Mead
3. Optimisation avec contraintes

2.3.1 Recherche linéaire

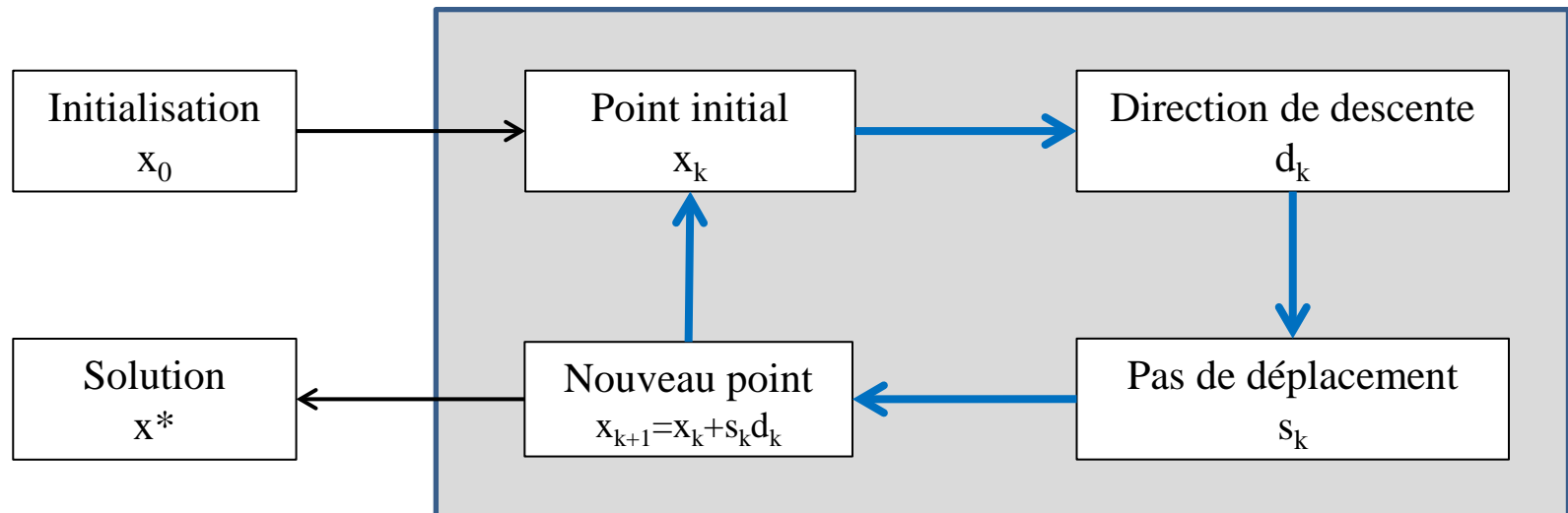
Problème sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x)$$

Etapes principales

A chaque itération

- Construction d'une direction de descente d_k à partir du point x_k
- Réglage du pas de déplacement s_k suivant d_k



2.3.1 Recherche linéaire

Etapes principales

A chaque itération

- Construction d'une direction de descente \mathbf{d}_k à partir du point \mathbf{x}_k
- Réglage du pas de déplacement s_k suivant \mathbf{d}_k

Direction de descente

\mathbf{d}_k est une direction de descente en \mathbf{x}_k si

$$\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$$

La direction de descente est construite à partir du gradient et du hessien.

- **Plus forte pente** → gradient (méthode d'ordre 1)
- **Préconditionnement** → hessien (méthode d'ordre 2)

Pas de déplacement

Le pas de déplacement s_k suivant \mathbf{d}_k doit vérifier

$$f(\mathbf{x}_k + s_k \mathbf{d}_k) < f(\mathbf{x}_k)$$

L'algorithme de recherche linéaire résout un problème de minimisation à une variable s

- Minimisation **exacte** → dichotomie (Fibonacci, nombre d'or, interpolation)
- Minimisation **approchée** → règles de pas acceptable (Armijo, Goldstein, Wolfe)

2.3.2 Direction de descente

- ☐ Plus forte pente
- ☐ Préconditionnement

2.3.2 Direction de descente

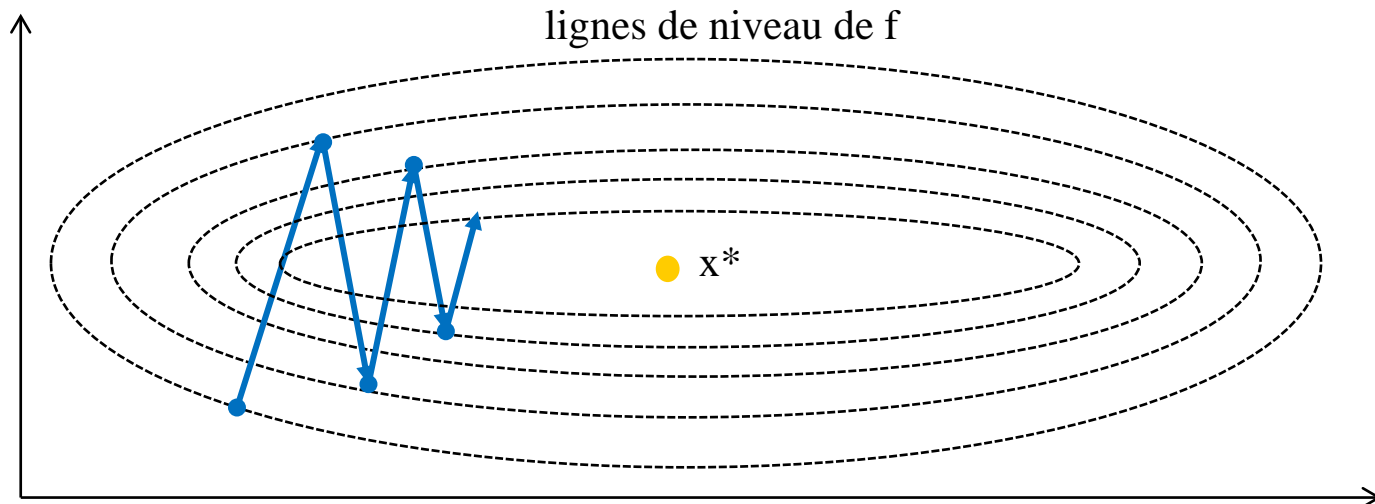
Plus forte pente

La direction de descente « naturelle » est celle du gradient = plus forte dérivée directionnelle

$$d_k = -\nabla f(x_k)$$

- Comportement caractéristique en zigzag
- **Convergence très lente** → méthode inefficace en général

Illustration



2.3.2 Direction de descente

Plus forte pente

Les directions successives de plus forte pente avec pas optimal sont **orthogonales**.

- Itération k

On cherche le minimum de f à partir de x_k suivant la direction $d_k = -\nabla f(x_k)$

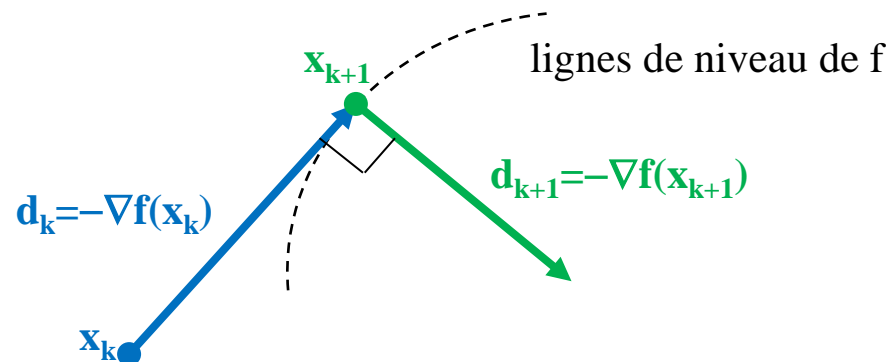
Le nouveau point est $x_{k+1} = x_k + sd_k$ avec le pas $s > 0$ solution de :

$$\min_{s \in \mathbb{R}} f(x_k + sd_k) \Rightarrow \frac{d}{ds} f(x_k + sd_k) = 0 \Rightarrow d_k^T \nabla f(x_k + sd_k) = 0 \Rightarrow d_k^T \nabla f(x_{k+1}) = 0$$

- Itération k+1

La direction de plus forte pente en x_{k+1} est $d_{k+1} = -\nabla f(x_{k+1}) \Rightarrow \boxed{d_k^T d_{k+1} = 0}$

Illustration



2.3.2 Exemple

Plus forte pente

- Fonction

$$f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$$

- Direction

$$d = -\nabla f(x) = \begin{pmatrix} -x_1 \\ -9x_2 \end{pmatrix}$$

- Pas

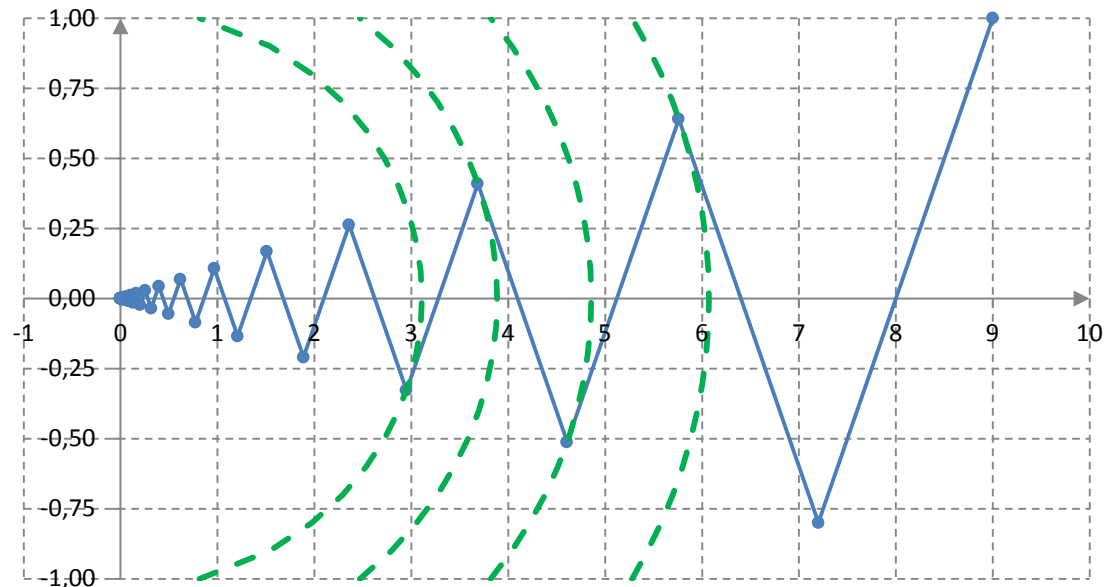
$$\min_s f(x + sd)$$

$$\rightarrow s = \frac{x_1^2 + 81x_2^2}{x_1^2 + 729x_2^2}$$

- Itération

$$x_{k+1} = x_k + s_k d_k$$

Iteration	x1	x2	f(x)	d1	d2	s	Erreur
0	9,000	1,000	45,000	-9,000	-9,000	0,2	9,055
1	7,200	-0,800	28,800	-7,200	7,200	0,2	7,244
2	5,760	0,640	18,432	-5,760	-5,760	0,2	5,795
3	4,608	-0,512	11,796	-4,608	4,608	0,2	4,636
4	3,686	0,410	7,550	-3,686	-3,686	0,2	3,709
5	2,949	-0,328	4,832	-2,949	2,949	0,2	2,967
10	0,966	0,107	0,519	-0,966	-0,966	0,2	0,972
20	0,104	0,012	0,006	-0,104	-0,104	0,2	0,104
30	1,11E-02	1,24E-03	6,90E-05	-1,11E-02	-1,11E-02	0,2	1,12E-02
40	1,20E-03	1,33E-04	7,95E-07	-1,20E-03	-1,20E-03	0,2	1,20E-03
50	1,28E-04	1,43E-05	9,17E-09	-1,28E-04	-1,28E-04	0,2	1,29E-04



2.3.2 Direction de descente

Préconditionnement

- On se donne une matrice H_k matrice **symétrique définie positive**.
 → factorisation de Cholesky de H_k : $H_k = L_k L_k^T$
- Changement de variable : $\tilde{x}_k = L_k^T x_k$
- Direction de plus forte pente pour : $\tilde{f}(\tilde{x}_k) = f(x_k) = f(L_k^{-T} \tilde{x}_k)$
 $\tilde{d}_k = -\nabla \tilde{f}(\tilde{x}_k) = -\nabla f(L_k^{-T} \tilde{x}_k) = -L_k^{-1} \nabla f(L_k^{-T} \tilde{x}_k) = -L_k^{-1} \nabla f(x_k) = -L_k^{-1} d_k$
- Itération en \tilde{x}_k : $\tilde{x}_{k+1} = \tilde{x}_k - s_k \nabla \tilde{f}(\tilde{x}_k)$
- Itération en x_k : $x_{k+1} = L_k^{-T} \tilde{x}_{k+1} = L_k^{-T} (\tilde{x}_k - s_k \nabla \tilde{f}(\tilde{x}_k)) = L_k^{-T} (L_k^T x_k - s_k L_k^{-1} d_k)$
 $\Rightarrow x_{k+1} = x_k - s_k L_k^{-T} L_k^{-1} d_k = x_k - s_k H_k^{-1} d_k$
- Le **préconditionnement par la matrice H_k** consiste à prendre comme direction de descente
 $d_k = -H_k^{-1} \nabla f(x_k)$
- On vérifie que d_k est une direction de descente
 $d_k^T \nabla f(x_k) = -\nabla f(x_k)^T H_k^{-1} \nabla f(x_k) < 0$ car H_k est définie positive

2.3.2 Direction de descente

Choix du préconditionnement

Toute matrice H_k **symétrique définie positive** convient.

Cas d'un hessien défini positif

- Si le hessien $\nabla^2 f(x_k)$ est défini positif, on peut prendre $H_k = \nabla^2 f(x_k)^{-1}$.

$$d_k = -H_k^{-1} \nabla f(x_k) = -\nabla^2 f(x_k)^{-1} \nabla f(x_k) \Rightarrow x_{k+1} = x_k - s_k \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

On obtient l'itération de Newton si le pas s_k vaut 1.

- On peut prendre pour H_k **l'approximation du hessien** donné par une méthode quasi-Newton.

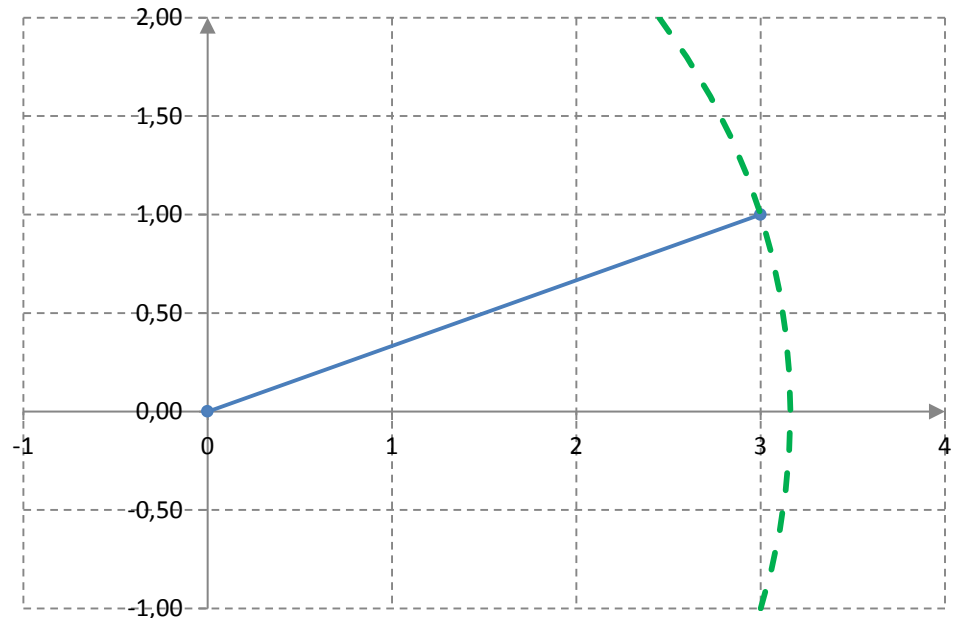
Cas d'un hessien non défini positif

- On peut effectuer la factorisation de Cholesky modifiée du hessien $\nabla^2 f(x_k)$ (ou son approximation quasi Newton) pour obtenir une matrice définie positive.
- On peut ajouter un multiple de l'identité : $H_k = (\nabla^2 f(x_k) + \tau I)^{-1}$ avec $\tau > 0$ assez grand
- On peut également prendre H_k diagonale : $(H_k)_i = \max \left(\varepsilon, \left(\frac{\partial^2 f}{\partial x_i^2}(x_k) \right)^{-1} \right)$, $\varepsilon > 0$
 à partir des dérivées secondes de f

2.3.2 Exemple

Préconditionnement

- Fonction : $f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2 \Rightarrow \nabla f(x) = \begin{pmatrix} x_1 \\ 9x_2 \end{pmatrix} \Rightarrow \nabla^2 f(x) = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}^T$
- Préconditionnement : $L = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \Rightarrow \begin{cases} \tilde{x}_1 = x_1 \\ \tilde{x}_2 = 3x_2 \end{cases} \Rightarrow \tilde{f}(\tilde{x}) = \frac{1}{2}\tilde{x}_1^2 + \frac{9}{2}\left(\frac{1}{3}x_2\right)^2 = \frac{1}{2}\tilde{x}_1^2 + \frac{1}{2}\tilde{x}_2^2$
- Direction : $\tilde{d} = -\nabla \tilde{f}(\tilde{x}) = \begin{pmatrix} -\tilde{x}_1 \\ -\tilde{x}_2 \end{pmatrix}$
- Pas : $\min_s \tilde{f}(\tilde{x} + s\tilde{d}) \rightarrow s = 1$
- Itération : $\tilde{x}_{k+1} = \tilde{x}_k + s_k \tilde{d}_k = 0$
 → **convergence en 1 itération**



2.3.3 Pas de déplacement

☐ Minimisation unidimensionnelle

☐ Minimisation exacte

- Méthode de dichotomie
- Méthode de Fibonacci
- Méthode du nombre d'or
- Interpolation quadratique

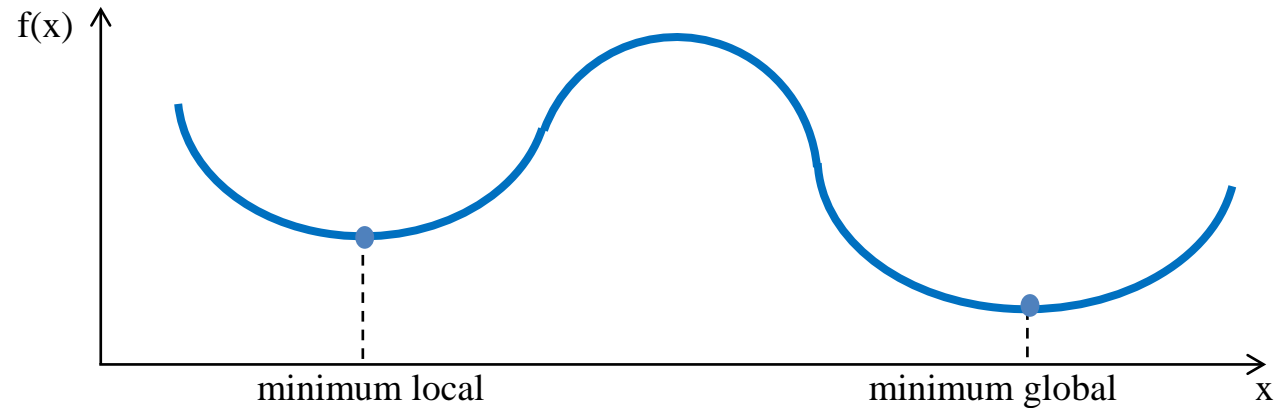
☐ Minimisation approchée

- Règle d'Armijo
- Règle de Goldstein
- Règle de Wolfe

2.3.3 Minimisation unidimensionnelle

Problème à une variable

$$\min_{x \in \mathbb{R}} f(x)$$



Méthodes

- **Minimisation exacte**

On cherche à trouver un minimum local x^* avec une précision donnée

→ réduction itérative de l'intervalle de recherche : dichotomie

→ réduction optimale : Fibonacci, nombre d'or

- **Minimisation approchée**

On cherche une réduction suffisante de la fonction sans déterminer précisément le minimum x^*

→ règles d'acceptation (Armijo, Goldstein, Wolfe)

→ limitation du nombre d'évaluations de la fonction

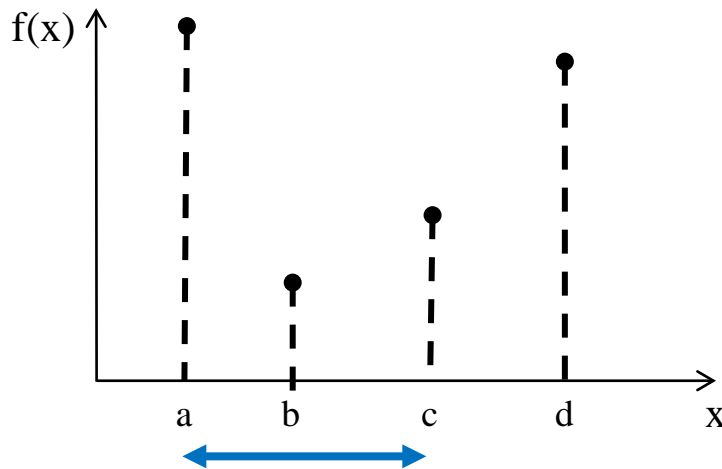
2.3.3 Minimisation exacte

Recherche par dichotomie

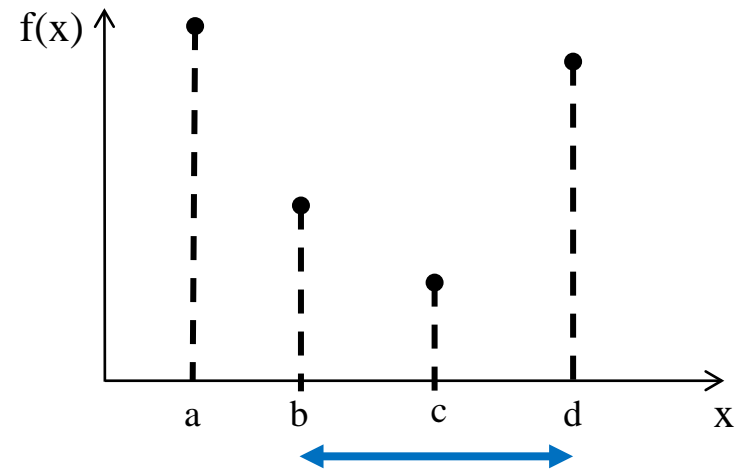
L'allure de la fonction f n'est pas connue.

On suppose qu'il existe un minimum unique dans l'intervalle de recherche $[a,d]$

- On évalue la fonction aux extrémités a et d : $\rightarrow f(a), f(d)$
- On évalue la fonction en 2 points b et c : $a < b < c < d$ $\rightarrow f(b), f(c)$
- On conserve soit l'intervalle $[a,c]$, soit l'intervalle $[b,d]$



On conserve $[a,c]$

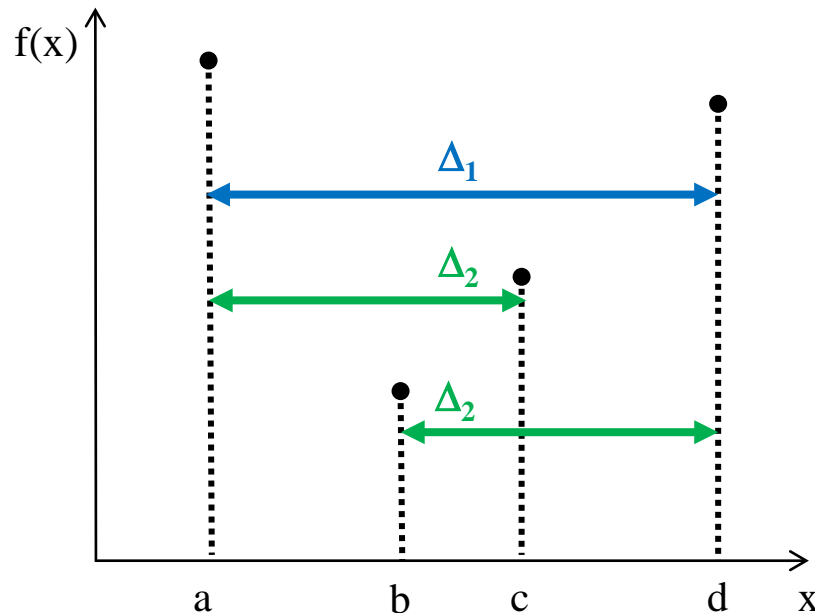


On conserve $[b,d]$

2.3.3 Minimisation exacte

Réduction optimale de l'intervalle

- On cherche b et c pour que l'intervalle restant soit le plus petit possible.
La taille de l'intervalle restant ne doit pas dépendre du choix de [a,c] ou [b,d].
- On note : Δ_1 la longueur de l'intervalle initial $\rightarrow \Delta_1 = d - a$
 Δ_2 la longueur de l'intervalle restant $\rightarrow \Delta_2 = c - a$ si on garde [a,c]
 ou $\Delta_2 = d - b$ si on garde [b,d]



$$\Delta_2 = d - b = c - a$$

$$\Rightarrow a + d = b + c \Rightarrow \frac{a + d}{2} = \frac{b + c}{2}$$



Les points b et c doivent être symétriques par rapport au milieu de l'intervalle [a,d].

2.3.3 Minimisation exacte

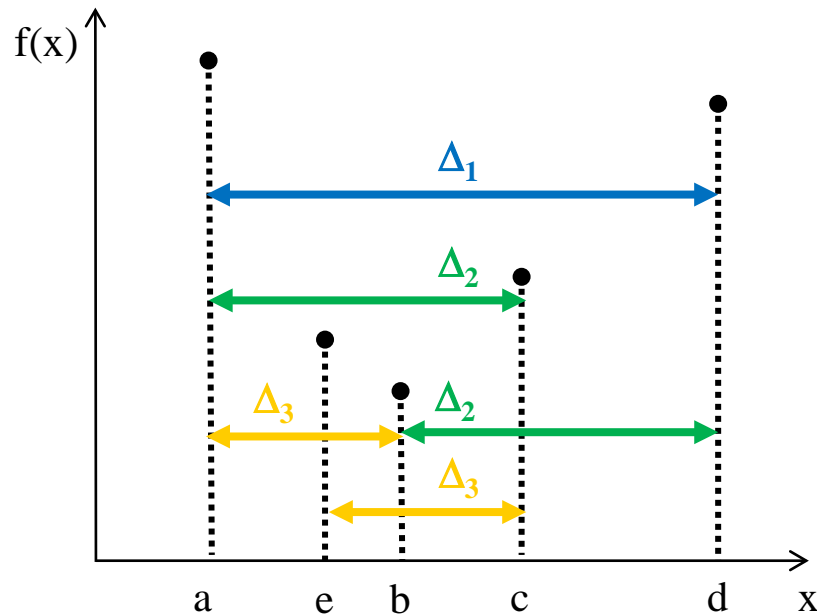
Réduction optimale de l'intervalle

- On suppose que l'intervalle restant est $[a,c]$.
- Pour **réutiliser le point b** à l'itération suivante, on choisit le nouveau point e symétrique de b par rapport au milieu de l'intervalle $[a,c]$.

$$\rightarrow \Delta_1 = d - a$$

$$\rightarrow \Delta_2 = c - a = d - b$$

$$\rightarrow \Delta_3 = b - a = c - e$$



$$d - a = (d - b) + (b - a)$$

$$\Rightarrow \Delta_1 = \Delta_2 + \Delta_3$$



Les longueurs Δ_k des intervalles successifs vérifient :

$$\Delta_k = \Delta_{k+1} + \Delta_{k+2}$$

2.3.3 Minimisation exacte

Réduction optimale de l'intervalle

- Les intervalles successifs sont de longueur Δ_k vérifiant : $\Delta_k = \Delta_{k+1} + \Delta_{k+2}$
- Après un nombre N d'itérations, on obtient un intervalle de longueur Δ_{N-1} .
- On définit la suite de nombres F_k par : $\Delta_k = F_{N-k} \Delta_{N-1}$ pour $k = 1, \dots, N-1$
- La suite $(F_n)_{n=1, \dots, N-1}$ vérifie

$$\Delta_k = \Delta_{k+1} + \Delta_{k+2} \Rightarrow \frac{\Delta_k}{\Delta_{N-1}} = \frac{\Delta_{k+1}}{\Delta_{N-1}} + \frac{\Delta_{k+2}}{\Delta_{N-1}} \Rightarrow F_{N-k} = F_{N-k-1} + F_{N-k-2}$$

$$\Rightarrow F_n = F_{n-1} + F_{n-2} \text{ pour } n = N-k, n = 3, 4, \dots, N-1$$

$$\Delta_{N-1} = F_1 \Delta_{N-1} \text{ pour } k = N-1 \Rightarrow F_1 = 1$$

- La suite est complètement déterminée par la valeur de F_2

$$\begin{cases} F_1 = 1 \\ F_2 \text{ à choisir} \\ F_n = F_{n-1} + F_{n-2} \text{ pour } n = 3, 4, \dots, N-1 \end{cases}$$

2.3.3 Minimisation exacte

Méthode de Fibonacci

- Pour un nombre N d'itérations fixé, on choisit F_2 pour que Δ_{N-1} soit minimal.

$$\Delta_{N-1} = \frac{\Delta_1}{F_{N-1}} \text{ minimal} \Rightarrow F_{N-1} \text{ maximal} \Rightarrow F_2 \text{ maximal}$$

- Valeur maximale de F_2 : $\Delta_{N-1} \geq \frac{1}{2} \Delta_{N-2} \left(\text{car } \Delta_k \geq \frac{1}{2} \Delta_{k-1} \right) \Rightarrow F_1 \geq \frac{1}{2} F_2 \Rightarrow F_{2\max} = 2$
- On obtient la **suite de Fibonacci** : 1, 2, 3, 5, 8, 13, 21, ...

Nombre d'itérations

La méthode de Fibonacci est optimale si l'on fixe à l'avance la précision requise sur la solution.

- Précision requise** : $\Delta_{N-1} = p \Rightarrow \Delta_{N-1} = \frac{\Delta_1}{F_{N-1}} \Rightarrow F_{N-1} = \frac{b-a}{p} \rightarrow \text{valeur de } N$
- Intervalle initial $[a,b]$: $\Delta_1 = b-a$

La méthode de Fibonacci donne le nombre minimal d'itérations N pour obtenir la solution **avec une précision donnée**.

- La disposition des premiers points b et c dépend de N : $\frac{\Delta_1}{\Delta_2} = \frac{F_{N-1}}{F_{N-2}} \rightarrow \Delta_2$

2.3.3 Minimisation exacte

Méthode du nombre d'or

La méthode de Fibonacci nécessite de changer la disposition des points à chaque itération.

La disposition des points n'est optimale que **pour une précision donnée**.

La méthode du nombre d'or est plus générale et plus simple à mettre en oeuvre.

- On impose un **rapport de réduction fixe** de l'intervalle à chaque itération.

$$\frac{\Delta_1}{\Delta_2} = \frac{\Delta_2}{\Delta_3} = \dots = \frac{\Delta_k}{\Delta_{k+1}} = \frac{\Delta_{k+1}}{\Delta_{k+2}} = \dots = \gamma \quad \text{avec} \quad \Delta_k = \Delta_{k+1} + \Delta_{k+2}$$

$$\Rightarrow \frac{\Delta_k}{\Delta_{k+1}} = 1 + \frac{\Delta_{k+2}}{\Delta_{k+1}} \Rightarrow \gamma = 1 + \frac{1}{\gamma} \Rightarrow \gamma^2 - \gamma - 1 = 0$$

- On obtient pour le rapport γ le nombre d'or :

→ **Méthode du nombre d'or** (ou de la **section dorée**)

$$\gamma = \frac{1 + \sqrt{5}}{2} \approx 1.618034$$

Optimalité

- La méthode du nombre d'or n'est pas optimale pour un nombre d'itérations donné N.

- Pour un nombre d'itérations grand : $\lim_{N \rightarrow \infty} \frac{F_N}{F_{N-1}} = \gamma$

→ La disposition des points de Fibonacci tend vers celle du nombre d'or.

2.3.3 Minimisation exacte

Comparaison Fibonacci - Nombre d'or

Le rapport de réduction de l'intervalle après n itérations vaut :

- Pour la méthode de Fibonacci : $\frac{\Delta_1}{\Delta_n} = F_n$
- Pour la méthode du nombre d'or : $\frac{\Delta_1}{\Delta_n} = \gamma^{n-1}$

Nombre d'itérations

Une itération de la méthode de Fibonacci ou du nombre d'or correspond à une évaluation de f.

Nombre d'évaluations

Rapport de réduction	Fibonacci	Nombre d'or
10^{-2}	11	13
10^{-3}	15	18
10^{-4}	20	22
10^{-5}	25	27
10^{-6}	30	31

2.3.3 Minimisation exacte

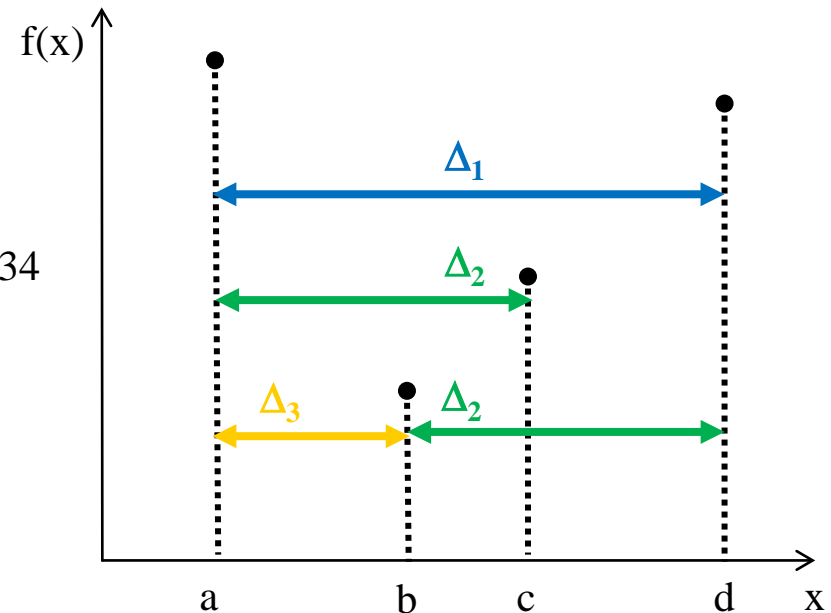
Méthode du nombre d'or

- Positionnement des points

$$\frac{\Delta_1}{\Delta_2} = \frac{\Delta_2}{\Delta_3} = \gamma = \frac{1+\sqrt{5}}{2} \approx 1.618034$$

$$\Rightarrow \begin{cases} \Delta_2 = \frac{1}{\gamma} \Delta_1 = r \Delta_1 \\ \Delta_3 = \frac{1}{\gamma^2} \Delta_1 = r^2 \Delta_1 \end{cases} \text{ avec } r = \frac{1}{\gamma} = \frac{\sqrt{5}-1}{2} \approx 0.618034$$

$$\Rightarrow \begin{cases} b = a + r^2 \Delta_1 \\ c = a + r \Delta_1 \\ d = a + \Delta_1 \end{cases}$$



- Itération

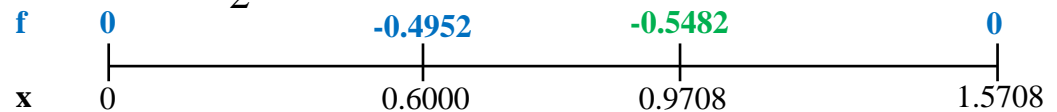
$$\text{Si } f(b) < f(c) \rightarrow \begin{cases} d \leftarrow c \\ c \leftarrow b \\ \Delta_1 \leftarrow \Delta_2 \\ b = a + r^2 \Delta_1 \end{cases} \quad \text{Si } f(b) > f(c) \rightarrow \begin{cases} a \leftarrow b \\ b \leftarrow c \\ \Delta_1 \leftarrow \Delta_2 \\ c = a + r \Delta_1 \end{cases}$$

2.3.3 Exemple

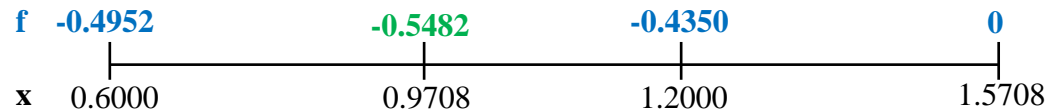
Méthode du nombre d'or

- Minimisation de $f(x) = -x \cos(x)$, $0 \leq x \leq \frac{\pi}{2}$

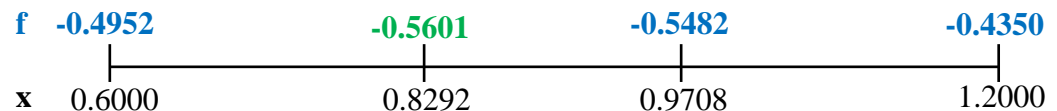
- Itération 1



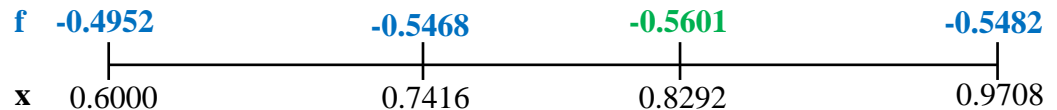
- Itération 2



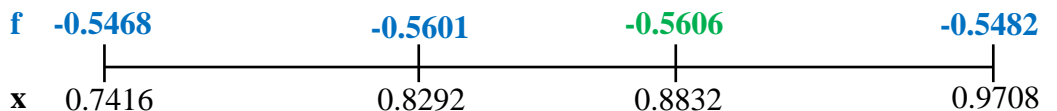
- Itération 3



- Itération 4



- Itération 5



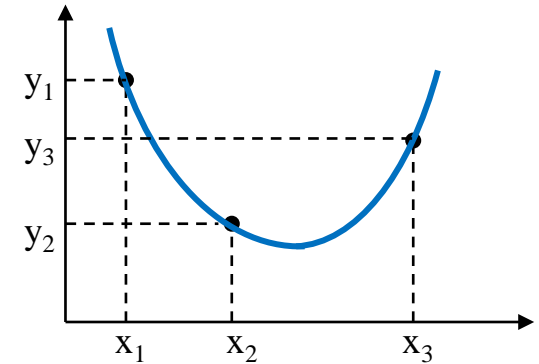
- Solution : $x^* \approx 0.8832 \rightarrow f(x^*) \approx -0.5606$
 au lieu de $x^* \approx 0.8603 \rightarrow f(x^*) \approx -0.5611$

2.3.3 Minimisation exacte

Interpolation quadratique

- On connaît la valeur de la fonction f en 3 points x_1, x_2, x_3 .

$$\begin{cases} y_1 = f(x_1) \\ y_2 = f(x_2) \\ y_3 = f(x_3) \end{cases}$$



- On construit le polynome q de degré 2 passant par les 3 points.

$$q(x) = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_3)(x - x_1)}{(x_2 - x_3)(x_2 - x_1)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} \rightarrow \begin{cases} q(x_1) = y_1 \\ q(x_2) = y_2 \\ q(x_3) = y_3 \end{cases}$$

- Dérivée du polynome q

$$q'(x) = y_1 \frac{(2x - x_2 - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(2x - x_3 - x_1)}{(x_2 - x_3)(x_2 - x_1)} + y_3 \frac{(2x - x_1 - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

On obtient une approximation du minimum de f en minimisant q .

2.3.3 Minimisation exacte

Interpolation quadratique

- La dérivée du polynome q s'écrit :

$$q'(x) = y_1 \frac{(2x - x_2 - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(2x - x_3 - x_1)}{(x_2 - x_3)(x_2 - x_1)} + y_3 \frac{(2x - x_1 - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

$$\Rightarrow q'(x) = \frac{-2x[y_1(x_2 - x_3) + y_2(x_3 - x_1) + y_3(x_1 - x_2)] + [y_1(x_2^2 - x_3^2) + y_2(x_3^2 - x_1^2) + y_3(x_1^2 - x_2^2)]}{(x_1 - x_2)(x_2 - x_3)(x_3 - x_1)}$$

$$\Rightarrow q'(x) = \frac{-2x(y_1 s_{23} + y_2 s_{31} + y_3 s_{12}) + (y_1 r_{23} + y_2 r_{31} + y_3 r_{12})}{(x_1 - x_2)(x_2 - x_3)(x_3 - x_1)} \quad \text{avec} \quad \begin{cases} s_{ij} = x_i - x_j \\ r_{ij} = x_i^2 - x_j^2 \end{cases}$$

- On cherche la valeur x_m qui annule la dérivée du polynome q .

$$q'(x_m) = 0 \Rightarrow 2x_m[y_1 s_{23} + y_2 s_{31} + y_3 s_{12}] + [y_1 r_{23} + y_2 r_{31} + y_3 r_{12}] = 0$$

$$\Rightarrow x_m = \frac{1}{2} \frac{y_1 r_{23} + y_2 r_{31} + y_3 r_{12}}{y_1 s_{23} + y_2 s_{31} + y_3 s_{12}} \quad \text{avec} \quad \begin{cases} s_{ij} = x_i - x_j \\ r_{ij} = x_i^2 - x_j^2 \end{cases}$$

2.3.3 Minimisation exacte

Interpolation quadratique

- On évalue la valeur de la fonction en $x_m \rightarrow y_m = f(x_m)$
 - \rightarrow On dispose de 4 points x_1, x_2, x_3, x_m avec les valeurs respectives de f : y_1, y_2, y_3, y_m
 - \rightarrow On conserve 3 points choisis parmi x_1, x_2, x_3, x_m selon :
 - la position de x_m par rapport à x_1, x_2, x_3
 - le minimum parmi y_m, y_1, y_2, y_3

Points retenus	$x_m < x_1$	$x_1 < x_m < x_2$	$x_2 < x_m < x_3$	$x_3 < x_m$
Minimum = y_m	(x_m, x_1, x_2)	(x_1, x_m, x_2)	(x_2, x_m, x_3)	(x_2, x_3, x_m)
Minimum = y_1	(x_m, x_1, x_2)	(x_1, x_m, x_2)	(x_1, x_2, x_m)	divergence
Minimum = y_2	divergence	(x_m, x_2, x_3)	(x_1, x_2, x_m)	divergence
Minimum = y_3	divergence	(x_m, x_2, x_3)	(x_2, x_m, x_3)	(x_2, x_3, x_m)

Cas de divergence : Le polynome est trop éloigné de la fonction
 Il faut un balayage plus fin avant l'interpolation quadratique.

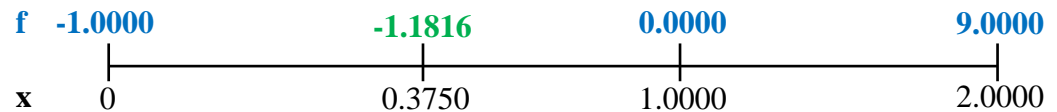
- On réitère l'interpolation quadratique avec les 3 nouveaux points jusqu'à réduire la taille de l'intervalle à la précision souhaitée.

2.3.3 Exemple

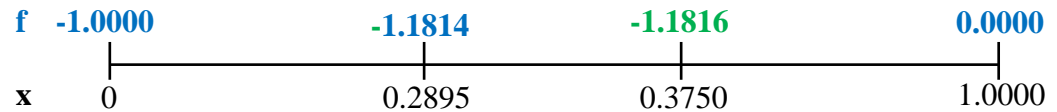
Interpolation quadratique

- Minimisation de $f(x) = (x-1)(x+1)^2$, $0 \leq x \leq 2$

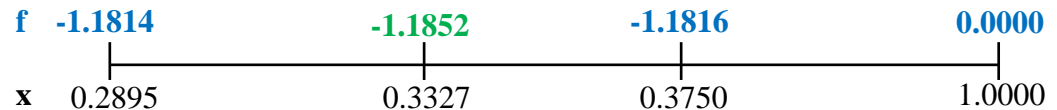
- Itération 1



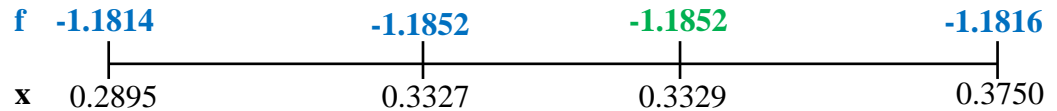
- Itération 2



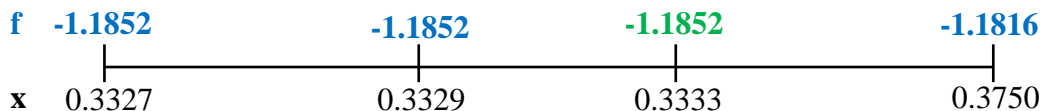
- Itération 3



- Itération 4



- Itération 5



- Solution : $x^* \approx 0.3333 \rightarrow f(x^*) \approx -1.1852$

2.3.3 Minimisation approchée

Principes

- Il n'est pas utile de réaliser une minimisation exacte suivant la direction de descente :
 - nécessite un grand nombre d'évaluations de la fonction
 - n'apporte pas une amélioration significative loin de la solution
- On peut se contenter d'une minimisation approchée
 - 2 **règles d'acceptation** d'un pas de déplacement

Notations

- x_k = point courant → $f(x_k)$
- d_k = direction de descente → $\nabla f(x_k)^T d_k < 0$
- Variation de la fonction f dans la direction d_k : $\varphi(s) = f(x_k + s d_k)$, $s \geq 0$

Règles d'acceptation du pas

- **Diminution suffisante** de la valeur de la fonction → condition d'Armijo
1^{ère} condition de Wolfe
- **Déplacement suffisant** par rapport au point initial → condition de Goldstein
2^{ème} condition de Wolfe

2.3.3 Minimisation approchée

Diminution suffisante

La fonction f doit décroître suffisamment pour accepter le pas.

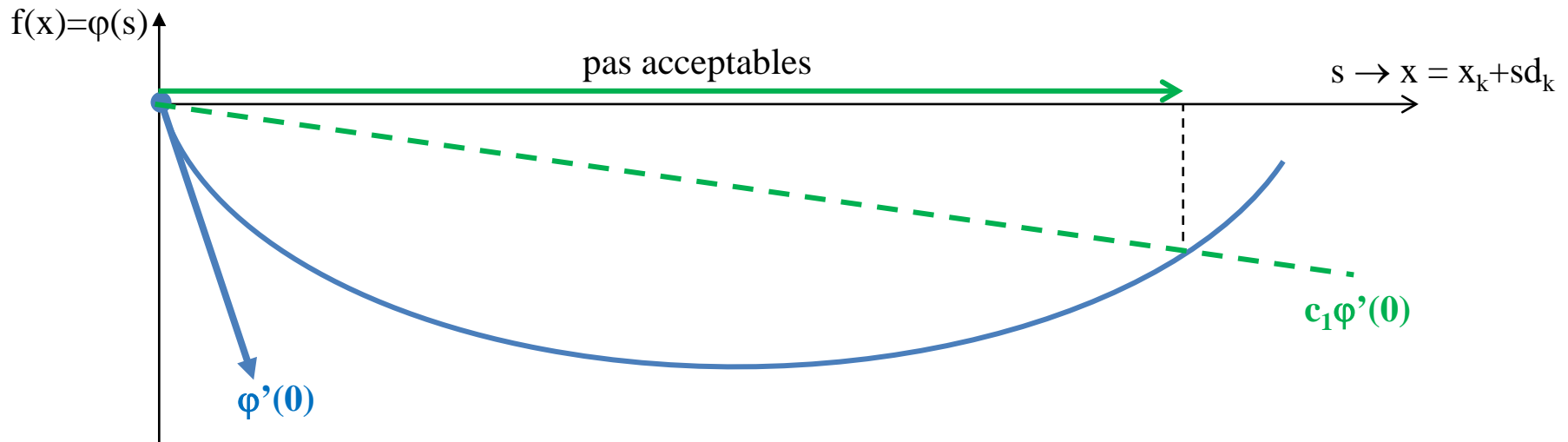
- Dérivée directionnelle de f suivant la direction d_k

$$\varphi(s) = f(x_k + sd_k), s \geq 0 \Rightarrow \varphi'(0) = \nabla f(x_k)^T d_k < 0$$

- On impose une diminution proportionnelle à la dérivée directionnelle, avec $0 < \varepsilon < c_1 < 0.5$

$$\varphi(s) < \varphi(0) + c_1 s \varphi'(0) \Leftrightarrow f(x_k + sd_k) < f(x_k) + c_1 s \nabla f(x_k)^T d_k \rightarrow \text{valeur typique } c_1 = 0.1$$

→ **Condition d'Armijo** ou 1^{ère} condition de Wolfe ou 1^{ère} condition de Goldstein



2.3.3 Minimisation approchée

Déplacement suffisant

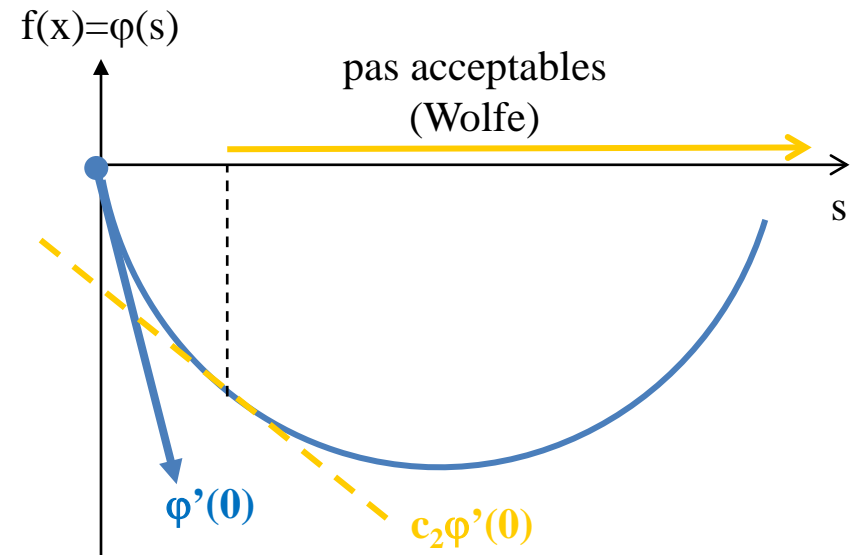
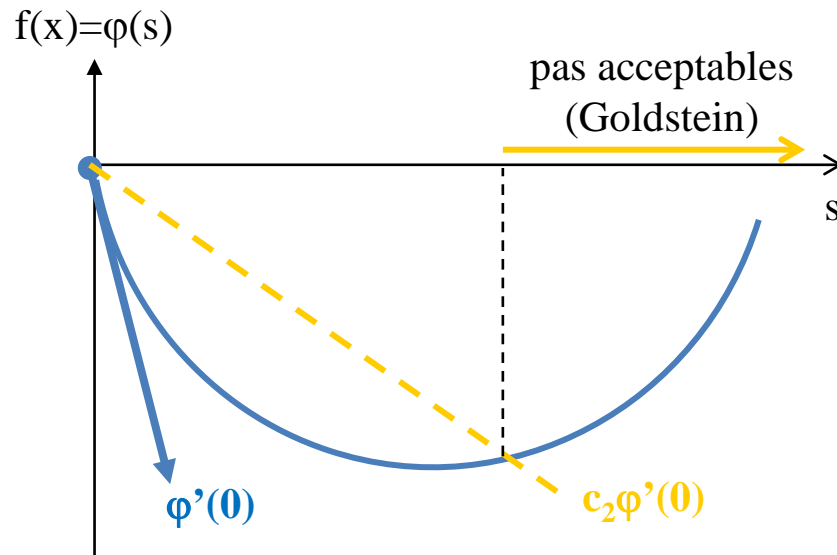
Le déplacement à partir du point initial doit être suffisant pour accepter le pas.

- Condition de Goldstein

$$\varphi(s) > \varphi(0) + c_2 s \varphi'(0) \Leftrightarrow f(x_k + s d_k) > f(x_k) + c_2 s \nabla f(x_k)^T d_k \rightarrow \text{valeur typique } c_2=0.9$$

- Condition de Wolfe : réduction de la dérivée

$$\varphi'(s) > c_2 \varphi'(0) \Leftrightarrow \nabla f(x_k + s d_k)^T d_k > c_2 \nabla f(x_k)^T d_k \rightarrow \text{valeur typique } c_2=0.9$$



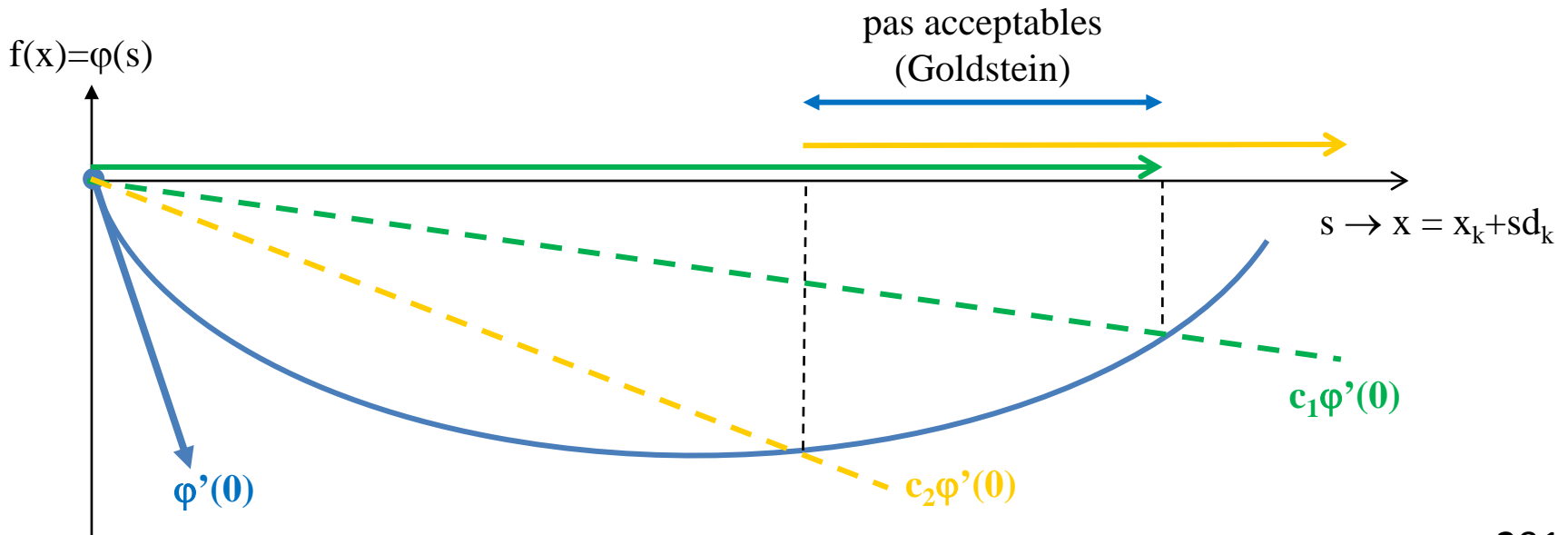
2.3.3 Minimisation approchée

Récapitulatif

- Conditions de Goldstein**
 (si la dérivée de f est coûteuse)

$$\begin{cases} \varphi(s) < \varphi(0) + c_1 s \varphi'(0) & \rightarrow c_1 \approx 0.1 \\ \varphi(s) > \varphi(0) + c_2 s \varphi'(0) & \rightarrow c_2 \approx 0.9 \end{cases}$$
- Conditions de Wolfe**
 (si la dérivée de f est disponible)

$$\begin{cases} \varphi(s) < \varphi(0) + c_1 s \varphi'(0) & \rightarrow c_1 \approx 0.1 \\ \varphi'(s) > c_2 \varphi'(0) & \rightarrow c_2 \approx 0.9 \end{cases}$$
 ou $|\varphi'(s)| < c_2 |\varphi'(0)| \rightarrow$ assure théoriquement la convergence



2.3.3 Réglage du pas

Méthode de dichotomie

On cherche un pas s vérifiant les conditions de Goldstein :
$$\begin{cases} \varphi(s) < \varphi(0) + c_1 s \varphi'(0) \\ \varphi(s) > \varphi(0) + c_2 s \varphi'(0) \end{cases}$$

Initialisation

- Intervalle initial : $[s_{\min}, s_{\max}]$ avec $s_{\min} = 0$
 s_{\max} assez grand
- Valeur initiale : $s=1$ (pas de Newton)

Itérations

Evaluation de $\varphi(s)$ et comparaison aux droites de Goldstein

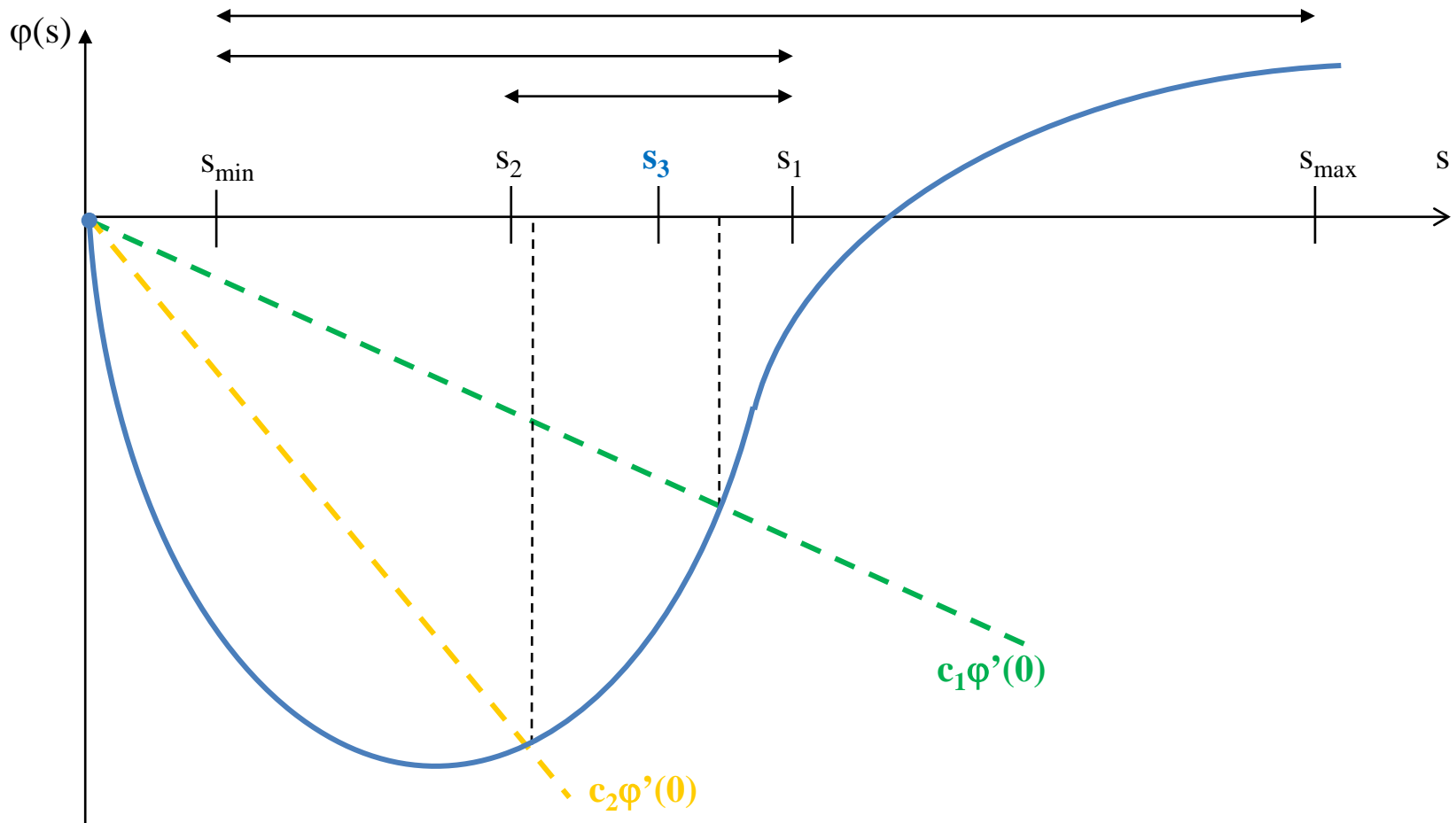
- Si s ne respecte pas la condition 1 \rightarrow amélioration insuffisante, pas trop grand : $s_{\max} = s$
- Si s ne respecte pas la condition 2 \rightarrow déplacement insuffisant, pas trop petit : $s_{\min} = s$
 \rightarrow Réduction de l'intervalle $[s_{\min}, s_{\max}]$
 \rightarrow Essai suivant : $s = \frac{1}{2}(s_{\min} + s_{\max})$

Arrêt

- Si s respecte les 2 conditions \rightarrow pas acceptable
- Si l'intervalle $[s_{\min}, s_{\max}]$ devient inférieur à un seuil donné

2.3.3 Réglage du pas

Illustration

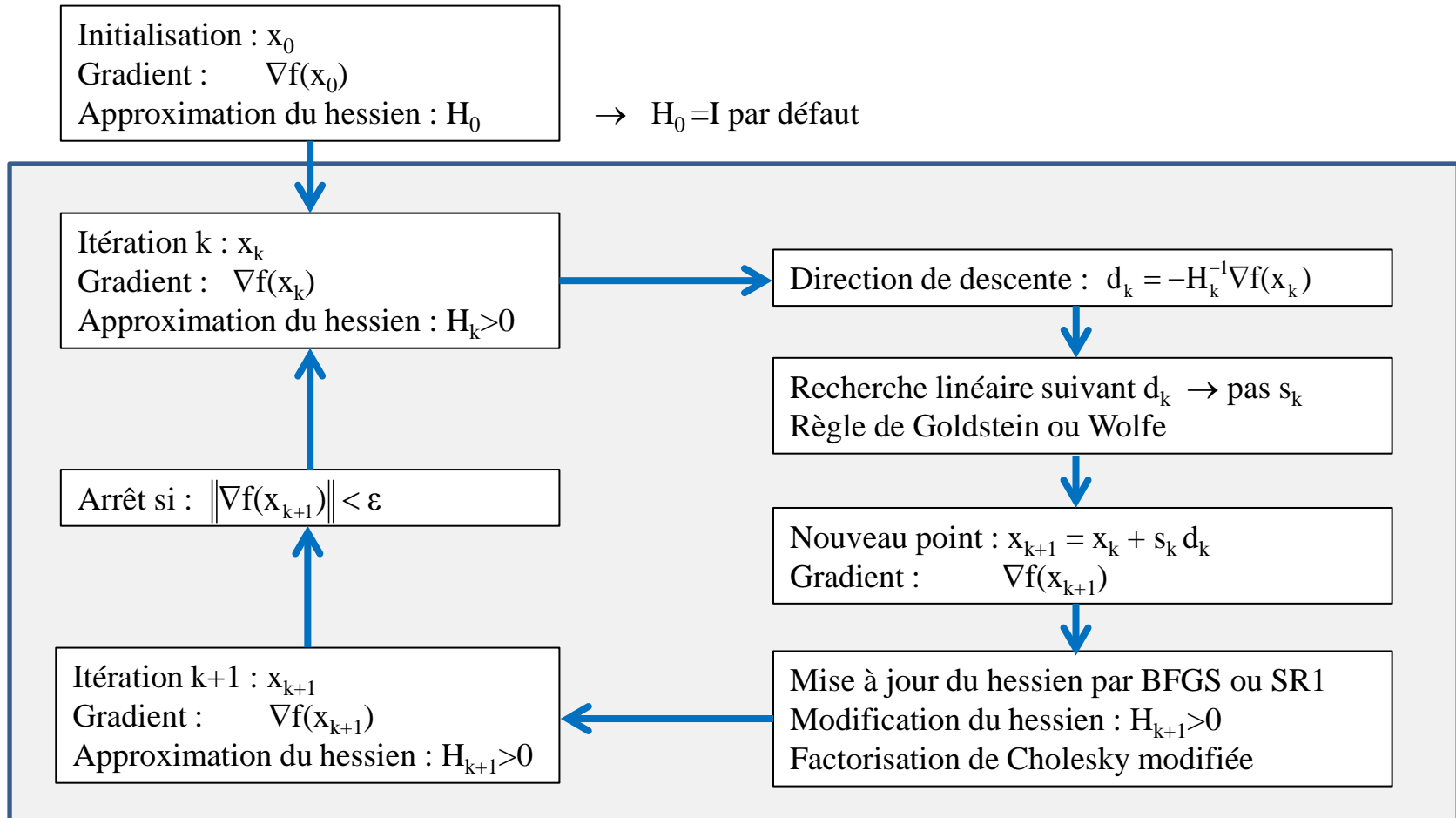


2.3.4 Algorithme

- ☐ Algorithme de recherche linéaire
- ☐ Convergence
- ☐ Exemple

2.3.4 Algorithme

Algorithme de recherche linéaire



2.3.4 Algorithme

Principaux résultats de convergence

- Si d est une direction de descente, et f est bornée inférieurement suivant d , alors il existe un pas s suivant d vérifiant les conditions de Wolfe
- Si les directions de descente ne deviennent pas « trop » orthogonales au gradient, l'algorithme de recherche linéaire avec les **conditions de Wolfe** est **globalement convergent**.
 $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0 \rightarrow \text{convergence vers un point stationnaire}$

En pratique

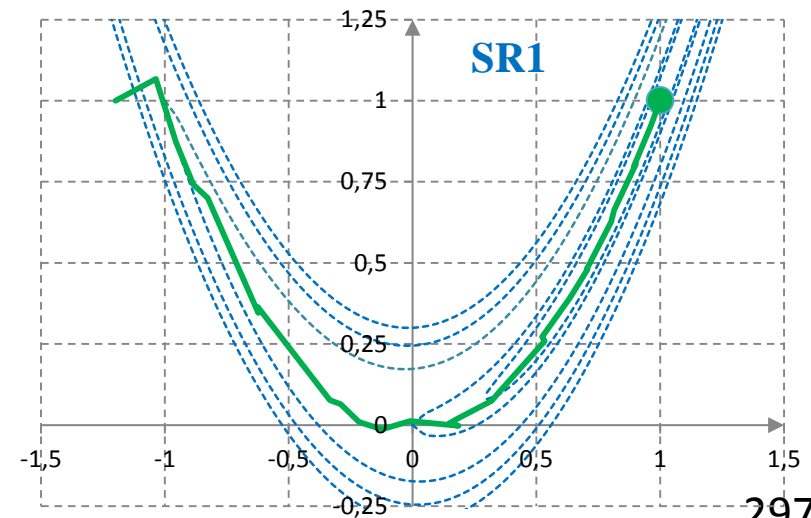
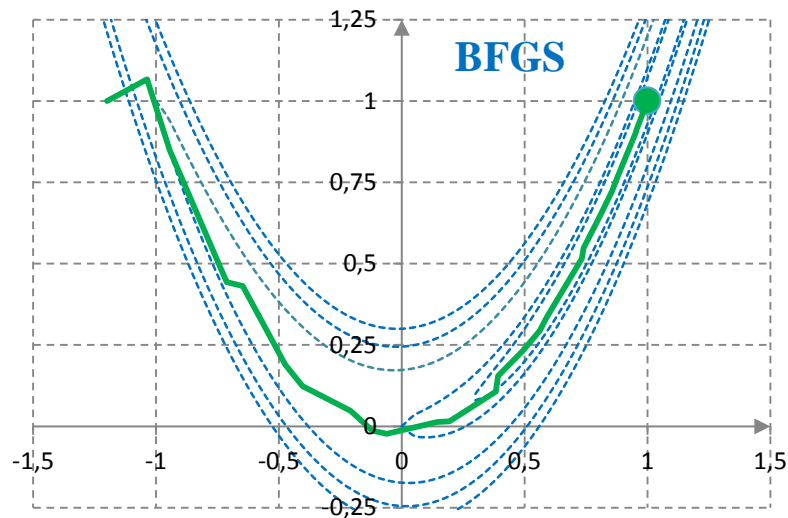
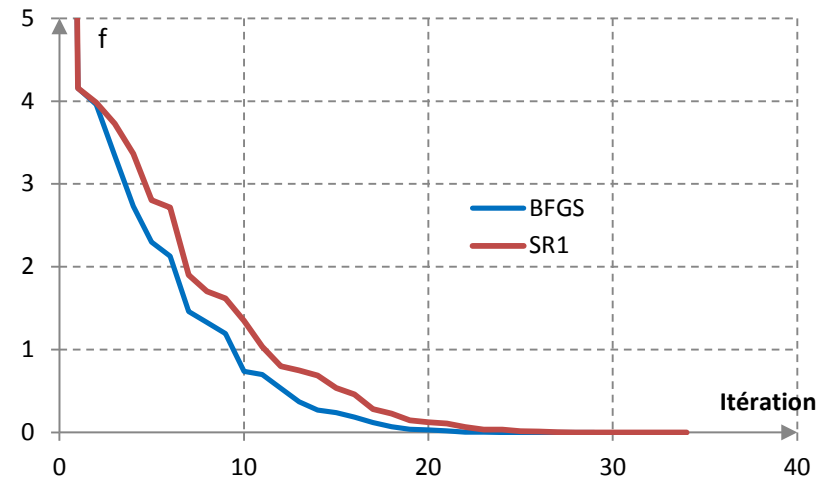
- Les directions de descente peuvent être construites avec BFGS ou SR1 (si matrices $H_k > 0$)
- La valeur des coefficients de Goldstein ou Wolfe c_1 et c_2 n'est pas critique pour la convergence.
- Le pas de Newton ($s=1$) est systématiquement testé, car il donne la solution si la fonction est proche de son modèle quadratique.
- La méthode de plus forte pente a la propriété de convergence globale, mais peut être très lente. On peut assurer la convergence globale d'un algorithme de recherche linéaire utilisant d'autres directions **en effectuant périodiquement une itération de plus forte pente**.

2.3.4 Exemple

Fonction de Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Point initial : $\begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$
- Recherche linéaire avec BFGS ou SR1



Sommaire

1. Bases théoriques
- 2. Optimisation sans contraintes**
 - 2.1 Méthodes de descente
 - 2.2 Méthode de Newton
 - 2.3 Recherche linéaire
 - 2.4 Région de confiance**
 - 2.4.1 Principes
 - 2.4.2 Modèle quadratique
 - 2.4.3 Rayon de confiance
 - 2.4.4 Algorithme
 - 2.5 Moindres carrés
 - 2.6 Méthode de Nelder-Mead
3. Optimisation avec contraintes

2.4.1 Région de confiance

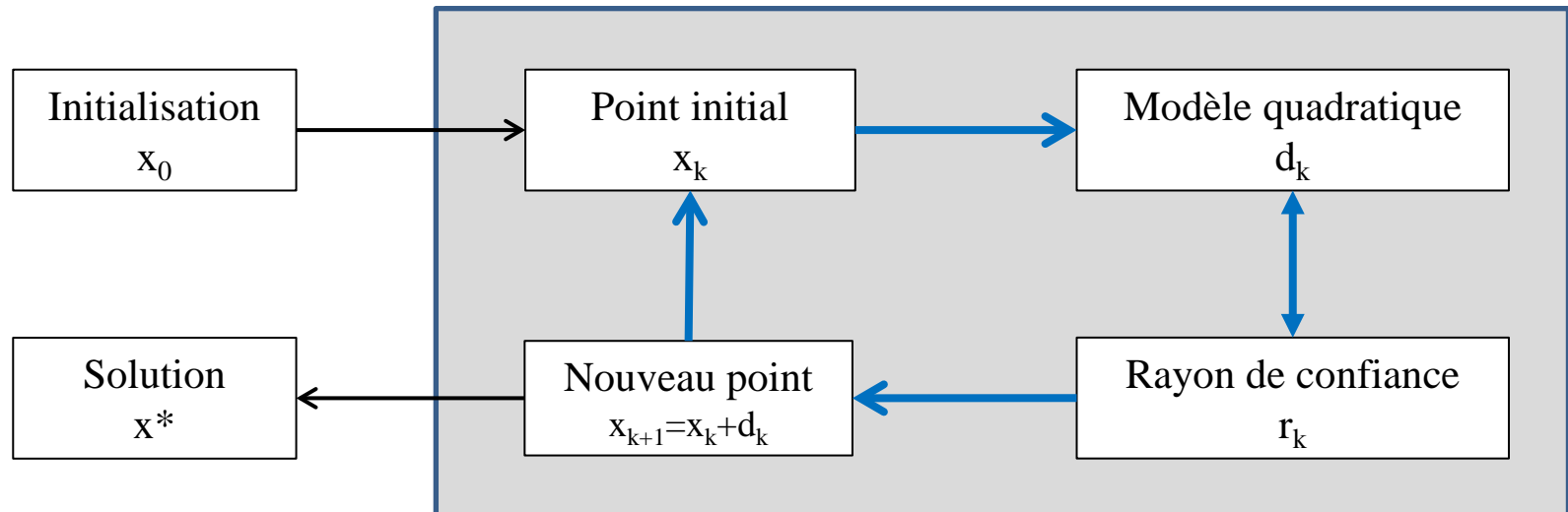
Problème sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x)$$

Etapes principales

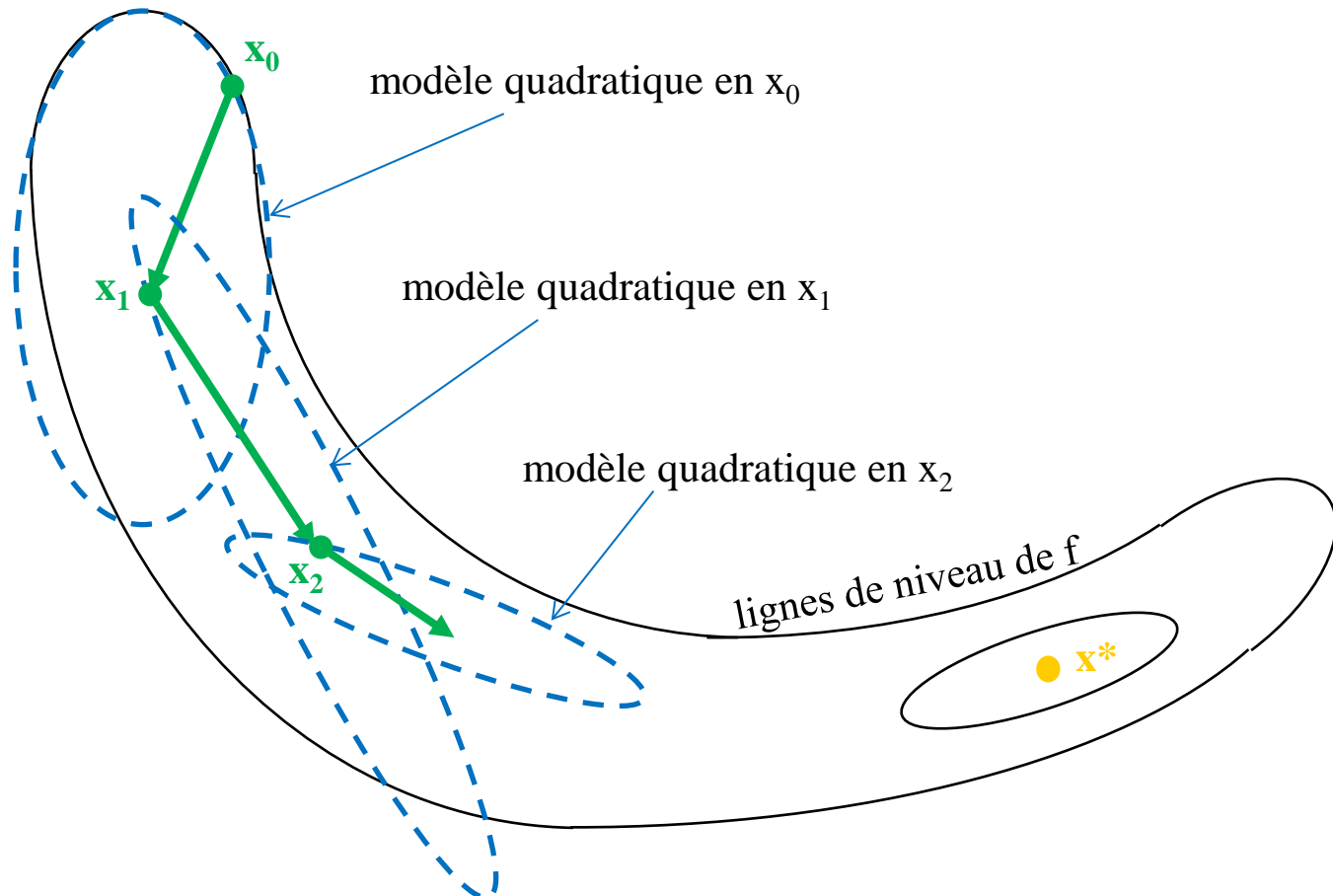
A chaque itération

- Résolution d'un modèle quadratique dans un rayon r_k autour du point $x_k \rightarrow$ déplacement d_k
- Réglage du rayon de confiance r_k pour obtenir une amélioration de f



2.4.1 Région de confiance

Illustration



2.4.2 Modèle quadratique

- ☐ Problème de région de confiance
- ☐ Conditions d'optimalité
- ☐ Résolution approchée
- ☐ Méthode dogleg

2.4.2 Problème de région de confiance

Modèle quadratique

- Au point courant x_k on approxime la fonction par son modèle quadratique.
 $d \in \mathbb{R}^n$ = déplacement à partir du point x_k
$$\hat{f}_k(x_k + d) = f(x_k) + d^T \nabla f(x_k) + \frac{1}{2} d^T \nabla^2 f(x_k) d$$
- Le modèle quadratique représente correctement la fonction au voisinage de x_k dans un rayon r_k
La région de confiance est le voisinage dans lequel l'approximation est « bonne » (à définir).

Région de confiance de rayon r_k : $\|d\| \leq r_k$
 r_k = rayon de confiance en x_k

Problème de région de confiance

- On cherche le minimum du modèle quadratique à l'intérieur de la région de confiance

$\min_{d \in \mathbb{R}^n} \hat{f}_k(x_k + d)$ sous $\|d\| \leq r_k$ → Problème de région de confiance

- On peut choisir différentes normes pour définir la région de confiance
 - **Norme 2** : région de confiance **circulaire**
 - **Norme ∞** : région de confiance **rectangulaire**

2.4.2 Problème de région de confiance

Notations

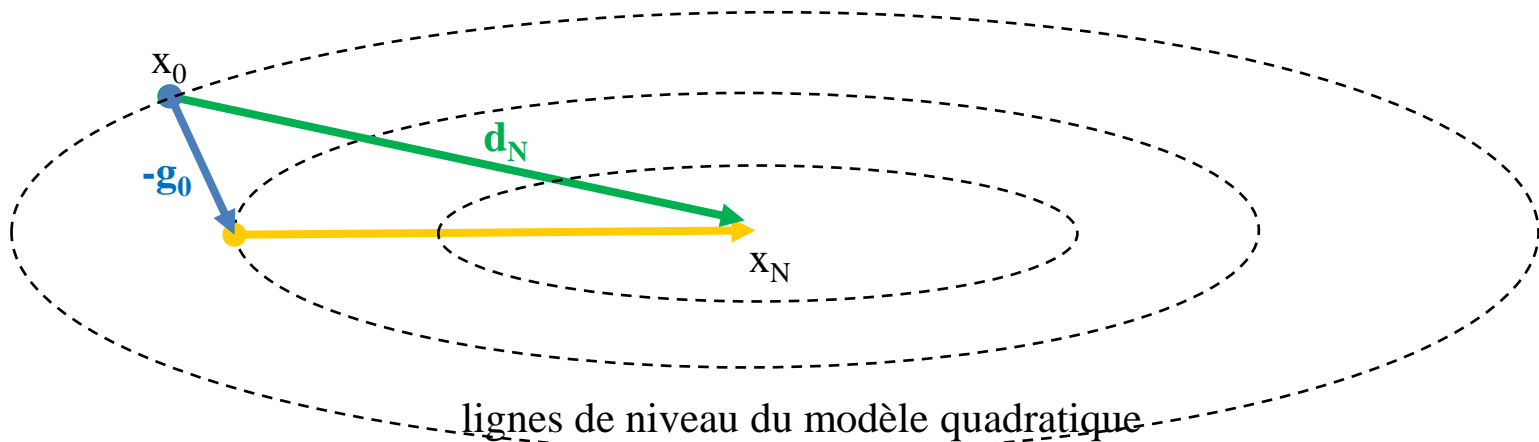
- Point courant : x_k noté x_0
- Fonction : $f(x_k)$ noté f_0
- Gradient : $\nabla f(x_k)$ noté g_0
- Hessien : $\nabla^2 f(x_k)$ noté H_0
- Déplacement à partir de x_0 : $d \in \mathbb{R}^n$

- Fonction modèle : $\hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d$

- Région de confiance : $\|d\| \leq r$

- Problème de région de confiance :

$$\min_{d \in \mathbb{R}^n} \hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d \text{ sous } \|d\| \leq r$$



2.4.2 Conditions d'optimalité

Région de confiance en norme 2

Le problème de région de confiance s'écrit avec la norme 2

$$\min_{d \in \mathbb{R}^n} \hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d \text{ sous } \frac{1}{2} (\|d\|^2 - r^2) \leq 0$$

- Lagrangien :
$$L(d, \mu) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d + \frac{1}{2} \mu (\|d\|^2 - r^2)$$
- Conditions d'ordre 1 :
$$\begin{cases} \nabla_d L(d^*, \mu^*) = g_0 + H_0 d^* + \mu^* d^* = 0 \\ \|d^*\| \leq r, \quad \mu^* \geq 0 \\ \mu^* (\|d^*\|^2 - r^2) = 0 \Leftrightarrow \mu^* (\|d^*\| - r) = 0 \end{cases}$$

Contrainte de région de confiance

- Si la contrainte de région de confiance est inactive, la contrainte peut être ignorée.
 La solution du problème quadratique sans contrainte est le **point de Newton**.
- Si la contrainte de région de confiance est active, la solution est au bord de la région de confiance : $\|d^*\| = r \Rightarrow (H_0 + \mu^* I) d^* = -g_0$
 On peut alors montrer que $H_0 + \mu^* I$ est semi-définie positive.
 Le problème quadratique est résolu avec la méthode de Newton et la matrice $H_0 + \mu^* I \geq 0$

2.4.2 Résolution approchée

Résolution du problème quadratique

- On doit résoudre à chaque itération le problème : $\min_{d \in \mathbb{R}^n} \hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d$ sous $\|d\| \leq r$
- Le problème doit éventuellement être résolu pour plusieurs valeurs du rayon.
 - résolution coûteuse si la contrainte est active
 - **solution approximative** par une méthode simplifiée (méthode dogleg)

Point de Newton

Le **point de Newton** x_N est la solution du problème quadratique sans contrainte.

$$\min_{d \in \mathbb{R}^n} \hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d \quad \Rightarrow \quad d_N = -H_0^{-1} g_0 \quad \Rightarrow \quad x_N = x_0 + d_N$$

Point de Cauchy

Le **point de Cauchy** x_C minimise le critère quadratique suivant le gradient : $d_C = -s g_0$, $s \geq 0$

$$\begin{aligned} \min_{s \in \mathbb{R}} \hat{f}(-s g_0) &= f_0 - s g_0^T g_0 + \frac{1}{2} s^2 g_0^T H_0 g_0 \quad \Rightarrow \quad s_C = \frac{g_0^T g_0}{g_0^T H_0 g_0} \\ &\Rightarrow \quad d_C = -s_C g_0 \quad \Rightarrow \quad x_C = x_0 + d_C \end{aligned}$$

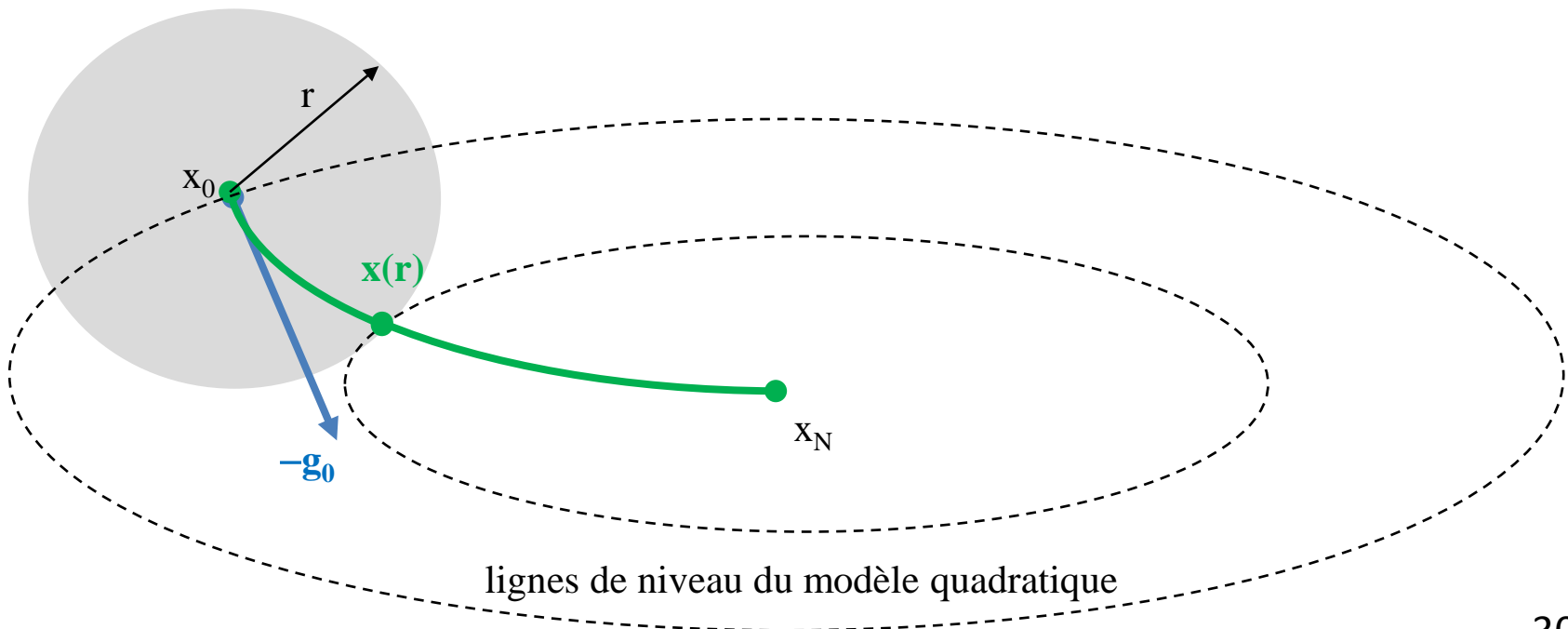
2.4.2 Résolution approchée

Résolution du problème quadratique

$$\min_{d \in \mathbb{R}^n} \hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d \quad \text{sous } \|d\| \leq r \quad \rightarrow \quad \text{solution } d(r) \in \mathbb{R}^n, x(r) = x_0 + d(r)$$

Lorsque le rayon de confiance r varie, la solution suit une courbe $x(r)$ allant de x_0 à x_N .

- $r = 0 \quad \rightarrow \quad x(0) = x_0 \quad (\text{tangente} = -g_0)$
- $r > |d_N| \quad \rightarrow \quad x(r) = x_N$



2.4.2 Méthode dogleg

Chemin dogleg

Le chemin dogleg est composé de 2 segments :

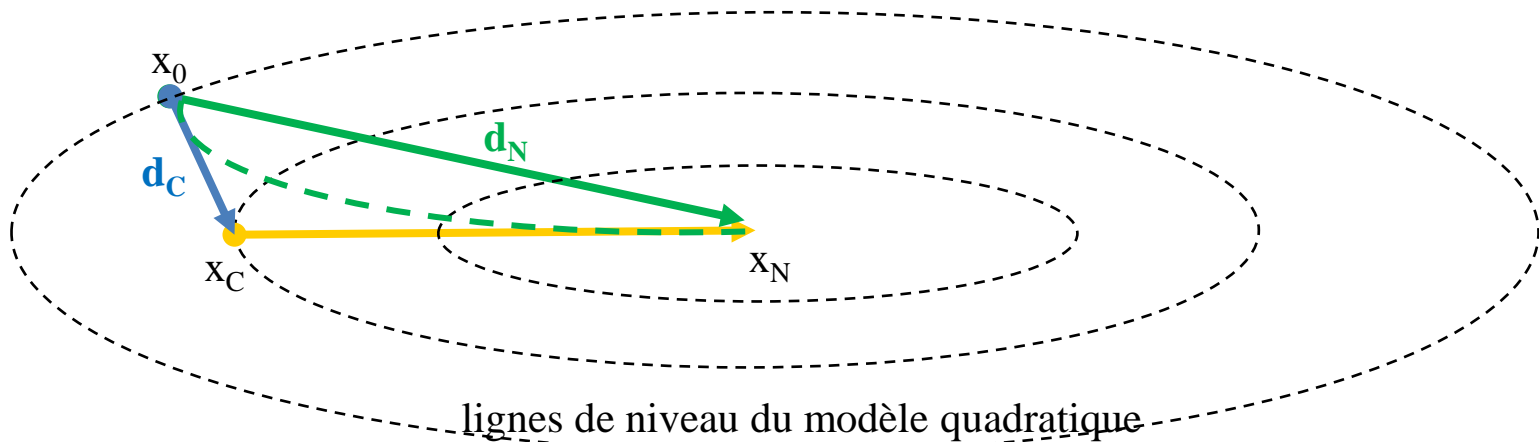
- Le **segment 1** joignant le point initial x_0 au point de Cauchy $x_C = x_0 + d_C$
- Le **segment 2** joignant le point de Cauchy x_C au point de Newton $x_N = x_C + d_N$

On restreint la recherche de la solution au chemin dogleg.

Paramétrage du chemin dogleg

Le chemin dogleg est paramétré par s : $x(s) = x_0 + d(s)$, pour $s \in [0,2]$

- Segment 1 $\rightarrow d(s) = s d_C$ pour $s \in [0,1[$
- Segment 2 $\rightarrow d(s) = d_C + (s-1)(d_N - d_C)$ pour $s \in [1,2]$



2.4.2 Méthode dogleg

Résolution du problème quadratique

On cherche la solution du problème quadratique sous contrainte de région de confiance

$$\min_{d \in \mathbb{R}^n} \hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d \quad \text{sous } \|d\| \leq r$$

Le point de Newton est la solution du problème quadratique sans contrainte \rightarrow 2 cas possibles

- Si le point de Newton est à l'intérieur de la région de confiance, la contrainte est **inactive**.

\rightarrow La solution est **d_N** .

- Sinon, la contrainte est **active** et peut être formulée comme une contrainte égalité.

$$\min_{d \in \mathbb{R}^n} \hat{f}(d) = f_0 + d^T g_0 + \frac{1}{2} d^T H_0 d \quad \text{sous } \|d\| = r$$

On restreint la recherche de la solution au chemin dogleg paramétré par $s \in [0, 2]$.

Segment 1 $\rightarrow d(s) = s d_C$ pour $s \in [0, 1[$

Segment 2 $\rightarrow d(s) = d_C + (s-1)(d_N - d_C)$ pour $s \in [1, 2]$

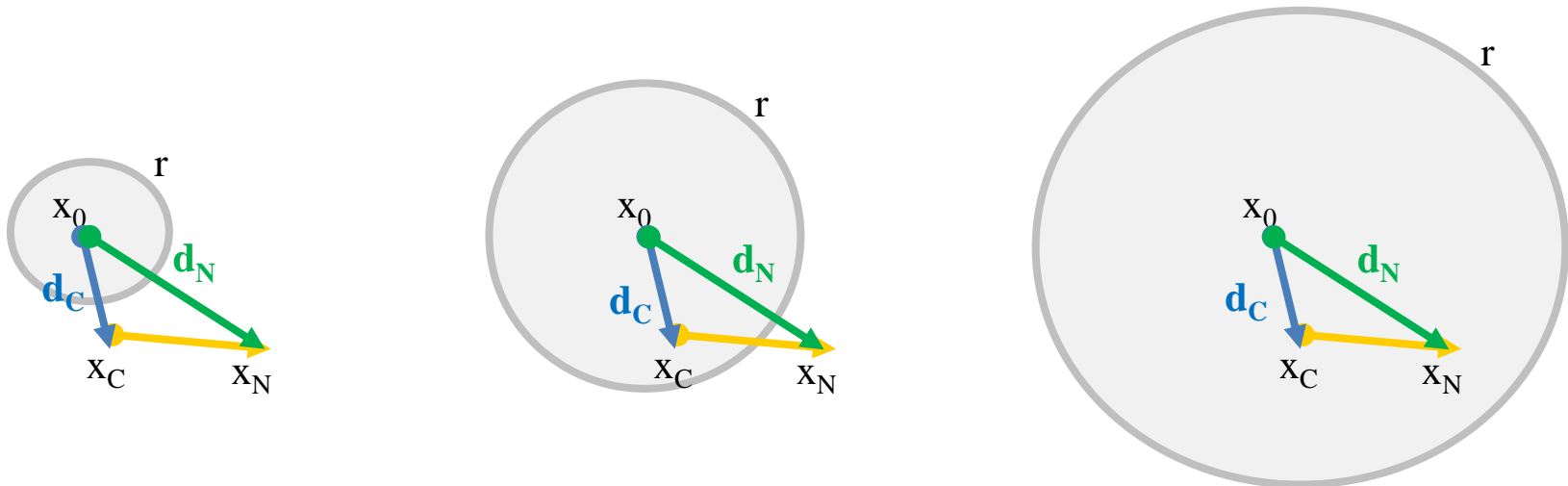
\rightarrow Il suffit de trouver l'intersection du chemin dogleg avec la région de confiance en résolvant **$|d(s)| = r$** .

2.4.2 Méthode dogleg

Position des points dogleg

Trois situations sont possibles par rapport au rayon de confiance.

- $|d_C| \geq r$ → le point de Cauchy est à l'extérieur de la région de confiance
- $|d_C| < r < |d_N|$ → le point de Cauchy est à l'intérieur de la région de confiance
le point de Newton est à l'extérieur de la région de confiance
- $|d_N| \leq r$ → le point de Newton est à l'intérieur de la région de confiance



2.4.2 Méthode dogleg

Solution du problème dogleg

La fonction $|d(s)|$ est croissante. La solution s^* de $|d(s)| = r$ est :

- Sur le segment 1 si $|d_C| \geq r$
- Sur le segment 2 si $|d_C| < r$

Solution sur le segment 1

$d(s) = s d_C$ avec $s \in [0, 1[$

$$\|s^* d_C\| = r \Rightarrow s^* = \frac{r}{\|d_C\|} \Rightarrow d^* = r \frac{d_C}{\|d_C\|} = -r \frac{g_0}{\|g_0\|} \quad \text{car } d_C // -g_0$$

Solution sur le segment 2

$d(s) = d_C + (s-1)(d_N - d_C)$ avec $s \in [1, 2]$

$$\|d_C + (s^*-1)(d_N - d_C)\| = r \Rightarrow (d_C + (s^*-1)(d_N - d_C))^T (d_C + (s^*-1)(d_N - d_C)) = r^2$$

L'équation du second degré en s^* admet une racine positive.

$$\Rightarrow s^* = 1 + \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{avec} \quad \begin{cases} a = \|d_N - d_C\|^2 \\ b = 2d_C^T (d_N - d_C) \\ c = \|d_C\|^2 - r^2 \end{cases}$$

2.4.2 Exemple

Méthode dogleg

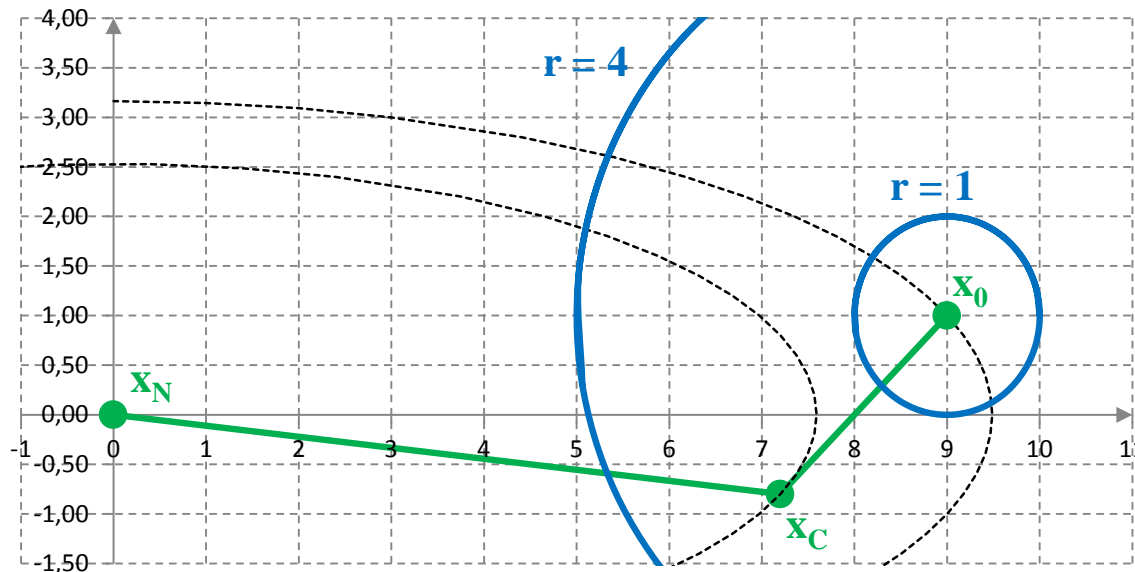
- Fonction $f(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$
- Point initial : $x_0 = \begin{pmatrix} 9 \\ 1 \end{pmatrix}$ $\nabla f(x_0) = \begin{pmatrix} 9 \\ 9 \end{pmatrix}$ $\nabla^2 f(x_0) = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}$
- Modèle quadratique : $\hat{f}(x_0 + p) = f(x_0) + \nabla f(x_0)^T p + \frac{1}{2} p^T \nabla^2 f(x_0) p = 45 + 9p_1 + 9p_2 + \frac{1}{2}p_1^2 + \frac{9}{2}p_2^2$
- Point de Cauchy : $\min_s \hat{f}(x_0 - s \nabla f(x_0)) \rightarrow \min_s 45 - 18(9s) + 5(9s)^2 \rightarrow s = \frac{1}{5} \Rightarrow x_C = \begin{pmatrix} 7.2 \\ -0.8 \end{pmatrix}$
- Point de Newton : $x_N = x_0 - (\nabla^2 f(x_0))^{-1} \nabla f(x_0) = \begin{pmatrix} 9 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1/9 \end{pmatrix} \begin{pmatrix} 9 \\ 9 \end{pmatrix} \Rightarrow x_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- Région de confiance de rayon r : $x_r = x_0 + r \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} = \begin{pmatrix} 9 + r \cos \theta \\ 1 + r \sin \theta \end{pmatrix} \rightarrow$ paramétrée par θ
- **Chemin dogleg** : segment 1 $\rightarrow x_{d1}(s) = x_0 + s(x_C - x_0) = \begin{pmatrix} 9 - 1.8s \\ 1 - 1.8s \end{pmatrix}$, $0 \leq s \leq 1$
 segment 2 $\rightarrow x_{d2}(s) = x_C + s(x_N - x_C) = \begin{pmatrix} 7.2 - 7.2s \\ -0.8 + 0.8s \end{pmatrix}$, $0 \leq s \leq 1$

2.4.2 Exemple

Méthode dogleg

$$f(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2 \quad x_0 = \begin{pmatrix} 9 \\ 1 \end{pmatrix} \Rightarrow x_C = \begin{pmatrix} 7.2 \\ -0.8 \end{pmatrix}, \quad x_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Région de confiance de rayon $r=1$: $\Rightarrow x_d \approx \begin{pmatrix} 8,293 \\ 0,293 \end{pmatrix}$ sur le **segment 1**
- Région de confiance de rayon $r=4$: $\Rightarrow x_d \approx \begin{pmatrix} 5,331 \\ -0,592 \end{pmatrix}$ sur le **segment 2**



— chemin dogleg

2.4.3 Rayon de confiance

- ☐ Rapport de réduction
- ☐ Réglage du rayon

2.4.3 Rapport de réduction

Validité du modèle quadratique

Le modèle quadratique est une approximation de la fonction valide dans un voisinage de x_0 .

→ Si le rayon de confiance est trop grand, le modèle quadratique n'est plus représentatif.

→ Il faut vérifier que la solution d^* du modèle quadratique donne l'amélioration attendue.

Rapport de réduction

- On définit le rapport de réduction ρ entre la variation prévue : $\hat{f}(x_0) - \hat{f}(x_0 + d^*)$
et la variation réalisée : $f(x_0) - f(x_0 + d^*)$

$$\text{Rapport de réduction : } \rho = \frac{f(x_0) - f(x_0 + d^*)}{\hat{f}(x_0) - \hat{f}(x_0 + d^*)}$$

- La valeur de ρ permet de vérifier si le modèle quadratique représente correctement la fonction, et d'adapter le rayon de confiance.
- $\rho \approx 1$ ou > 1 : amélioration réelle $>$ amélioration prévue → modèle bon
- $\rho \approx 0$ ou < 0 : amélioration réelle $<$ amélioration prévue → modèle mauvais

Le rayon est réglé de façon itérative en fonction de la valeur de ρ .

2.4.3 Réglage du rayon

Réglage du rayon

On modifie le rayon de confiance en fonction du rapport de réduction $\rho = \frac{f(x_0) - f(x_0 + d^*)}{\hat{f}(x_0) - \hat{f}(x_0 + d^*)}$

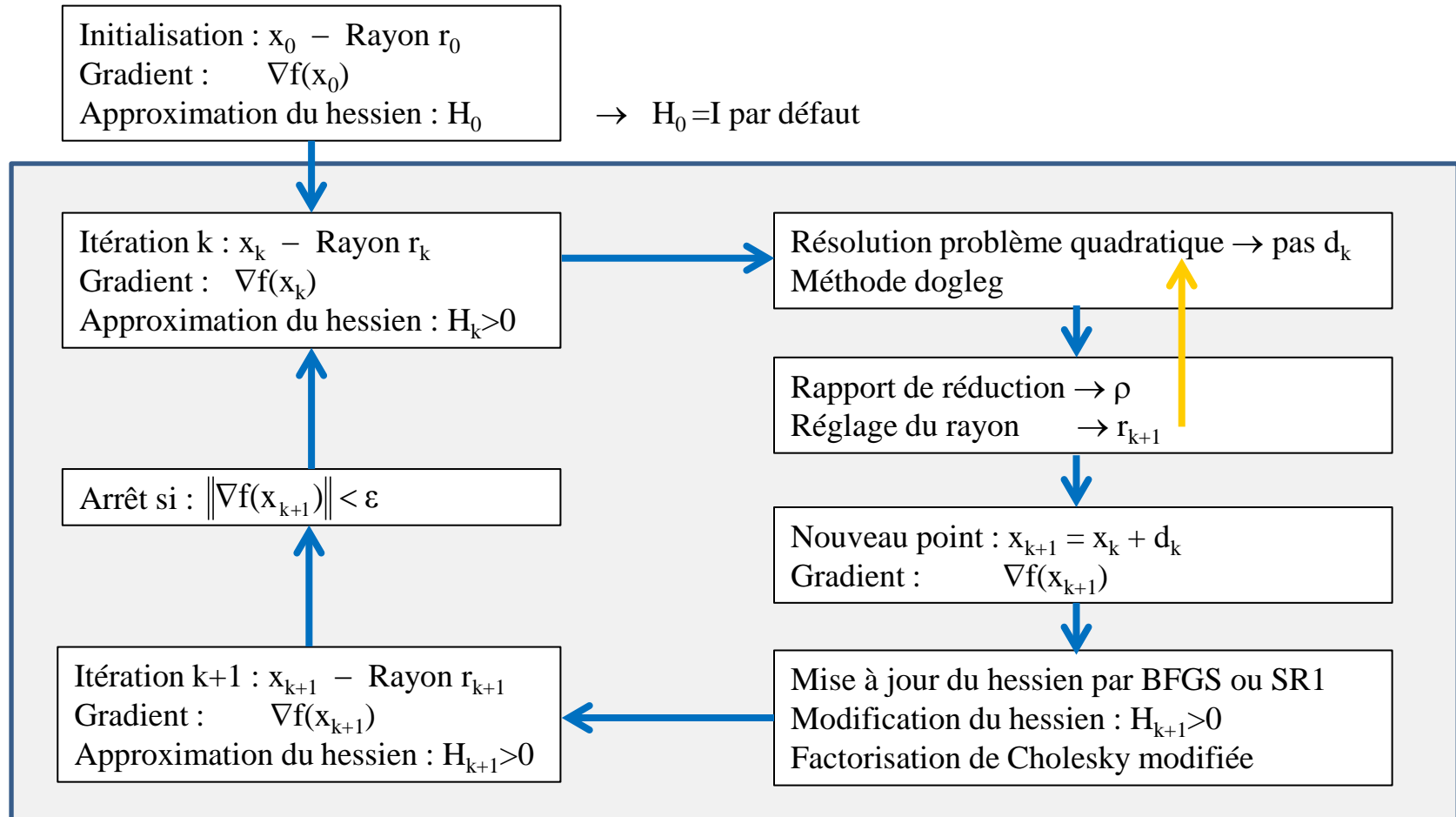
- $\rho > \rho_2$ avec $\rho_2 = 0.9$
La fonction f décroît au moins comme prévu : le modèle est bon.
 - On **accepte** le nouveau point.
 - On passe à l'itération suivante **en multipliant par 2** le rayon de confiance.
- $\rho < \rho_1$ avec $\rho_1 = 0.01$
La fonction f augmente au lieu de diminuer comme prévu : le modèle est mauvais.
 - On **rejette** le nouveau point.
 - On **reprend** l'itération **en divisant par 2** le rayon de confiance.
- $\rho_1 < \rho < \rho_2$
La fonction décroît, mais moins que prévu : le modèle est correct.
 - On **accepte** le nouveau point.
 - On passe à l'itération suivante **sans changer** le rayon de confiance.

2.4.4 Algorithme

- ❑ Algorithme de région de confiance
- ❑ Convergence
- ❑ Exemple
 - Région de confiance norme 2
 - Région de confiance norme ∞

2.4.4 Algorithme

Algorithme de région de confiance



2.4.4 Algorithme

Principaux résultats de convergence

- Lorsque le rayon de confiance est petit, la solution dogleg est suivant le gradient.
L'itération est équivalente à la **méthode de plus forte pente** → convergence lente
- Lorsque le rayon de confiance est grand, la solution dogleg est le point de Newton.
L'itération est équivalente à la **méthode de Newton** → convergence rapide

En pratique

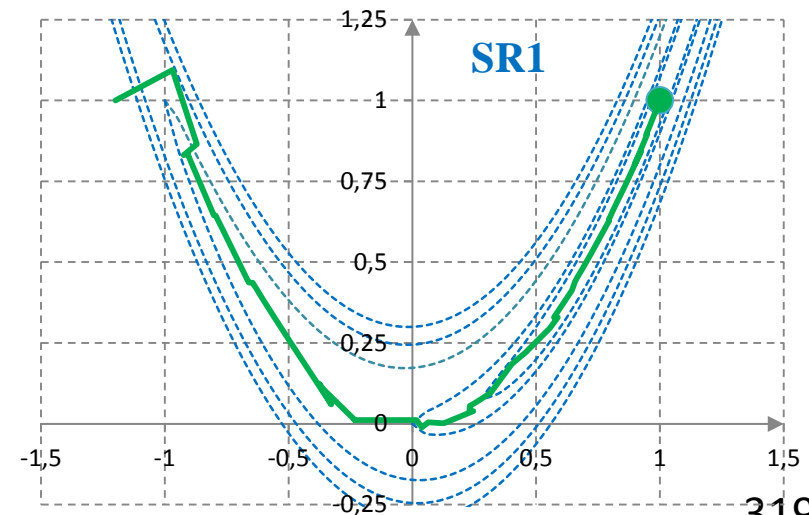
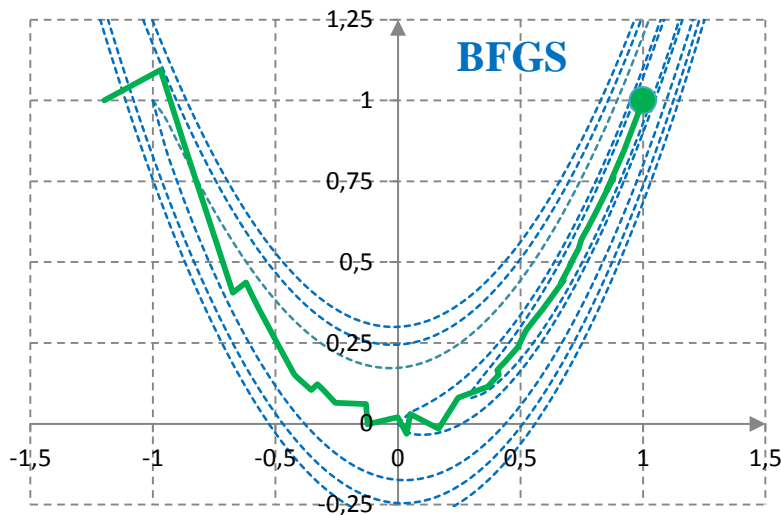
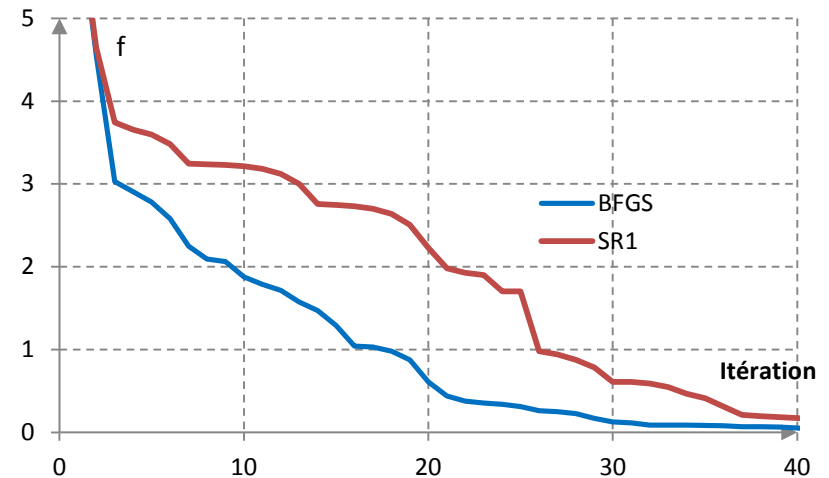
- On peut combiner l'algorithme de région de confiance avec BFGS ou SR1.
Il n'est pas nécessaire que le hessien soit défini positif.
Si le hessien n'est pas défini positif, la méthode SR1 est plus efficace.
- La valeur des seuils de réduction ρ_1 et ρ_2 pour régler le rayon n'est pas critique pour la convergence.
- Le point de Newton x_N est systématiquement testé,
car il donne la solution s'il est à l'intérieur de la région de confiance.
Sinon, on cherche une solution approchée sur le chemin dogleg.
- D'autres méthodes de résolution approchée du problème quadratique existent.

2.4.4 Exemple

Fonction de Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Point initial : $\begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$
- Région de confiance avec BFGS ou SR1
- **Norme 2** : région de confiance **circulaire**

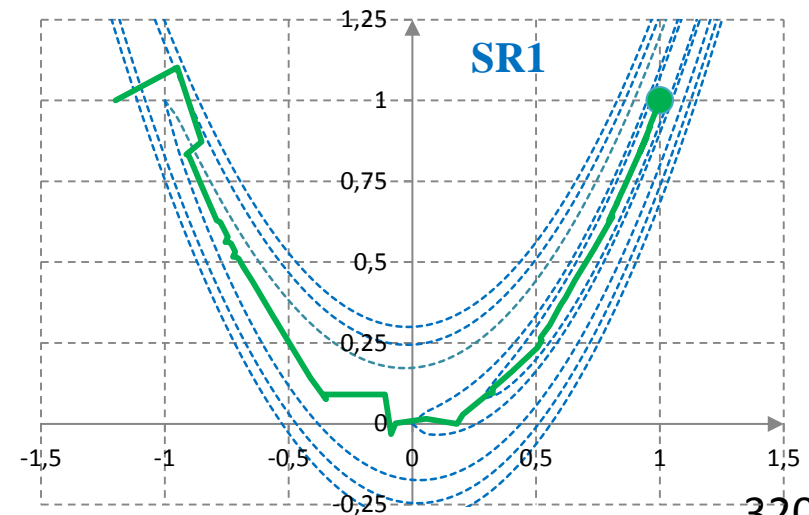
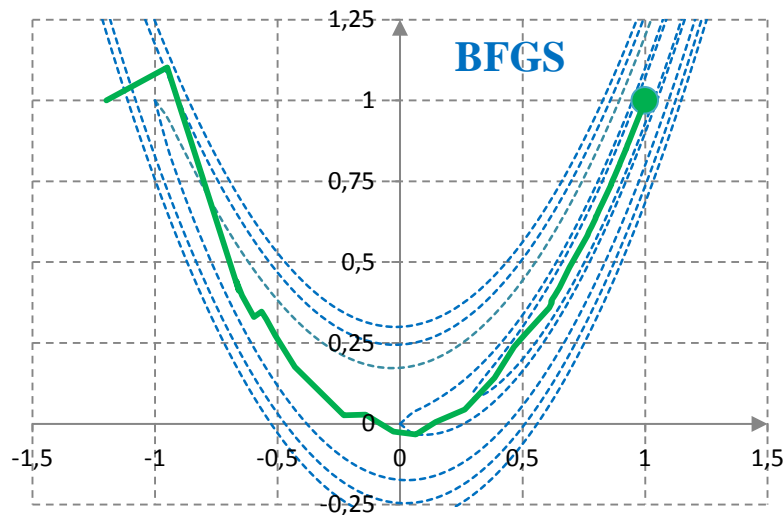
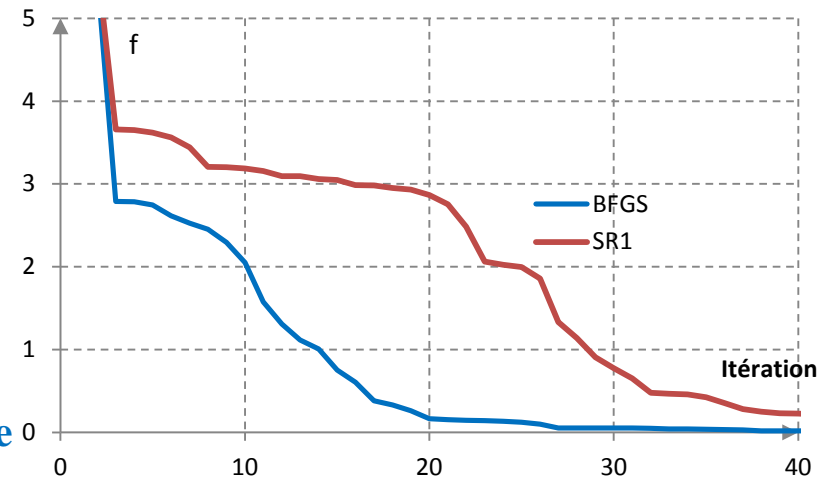


2.4.4 Exemple

Fonction de Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Point initial : $\begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$
- Région de confiance avec BFGS ou SR1
- **Norme ∞** : région de confiance **rectangulaire**



Sommaire

1. Bases théoriques
2. **Optimisation sans contraintes**
 - 2.1 Méthodes de descente
 - 2.2 Méthode de Newton
 - 2.3 Recherche linéaire
 - 2.4 Région de confiance
 - 2.5 **Moindres carrés**
 - 2.5.1 Formulation
 - 2.5.2 Méthode de Gauss-Newton
 - 2.5.3 Moindres carrés linéaires
 - 2.5.4 Filtre de Kalman
 - 2.5.5 Méthode du gradient conjugué
 - 2.6 Méthode de Nelder-Mead
3. Optimisation avec contraintes

2.5.1 Formulation

Problème de moindres carrés

$$\boxed{\min_{x \in \mathbb{R}^n} \frac{1}{2} \|r(x)\|^2} \rightarrow \text{problème (MC)}$$

- Variables $x \in \mathbb{R}^n$: n paramètres d'ajustement d'un modèle
- Résidus $r(x) \in \mathbb{R}^m$: m écarts entre modèle et mesures
- Critère $f(x) \in \mathbb{R}$: $f(x) = \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} r(x)^T r(x) = \frac{1}{2} \sum_{i=1}^m r_i(x)^2$
 \rightarrow minimisation d'écart quadratique
- Gradient : $\nabla f(x) = \nabla r(x) r(x) = \sum_{i=1}^m \nabla r_i(x) r_i(x)$ avec $\begin{cases} \nabla r(x) \in \mathbb{R}^{n \times m} \\ \nabla r_i(x) \in \mathbb{R}^{n \times 1} \end{cases}$
- Hessian : $\nabla^2 f(x) = \sum_{i=1}^m \left(\nabla r_i(x) \nabla r_i(x)^T + \nabla^2 r_i(x) r_i(x) \right)$
 $= \nabla r(x) \nabla r(x)^T + \sum_{i=1}^m \nabla^2 r_i(x) r_i(x)$
- Résolution : en appliquant la **méthode de Newton**

2.5.2 Méthode de Gauss-Newton

Méthode de Newton

L'itération de Newton au point x_k est : $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$ → déplacement d_k

Avec
$$\begin{cases} \nabla f(x) = \nabla r(x)r(x) = \sum_{i=1}^m \nabla r_i(x)r_i(x) \\ \nabla^2 f(x) = \nabla r(x)\nabla r(x)^T + \sum_{i=1}^m \nabla^2 r_i(x)r_i(x) \end{cases}$$

Méthode de Gauss-Newton

- L'évaluation du hessien nécessite les dérivées secondes des fonctions $r_i(x)$
 → calcul très coûteux si m est grand
- Pour accélérer les calculs, on ignore le second terme : $\nabla^2 f(x) \approx \nabla r(x)\nabla r(x)^T$
 Cette approximation du hessien n'utilise que les dérivées premières.
 La matrice obtenue est de plus toujours **semi-définie positive**
 → nécessaire pour la convergence de la méthode de Newton
- L'itération de **Gauss-Newton** au point x_k est : $\nabla r(x_k)\nabla r(x_k)^T d_k = -\nabla r(x_k)r(x_k)$

2.5.2 Méthode de Gauss-Newton

Méthode de Gauss-Newton

La méthode de Gauss-Newton équivaut à considérer un **modèle linéaire** des fonctions $r_i(x)$ en x_k

$$\hat{r}_i(x) = r_i(x_k) + \nabla r_i(x_k)^T (x - x_k), i = 1, \dots, m \quad \Leftrightarrow \quad \hat{r}(x) = r(x_k) + \nabla r(x_k)^T (x - x_k)$$

Le problème devient : $\min_{x \in \mathbb{R}^n} \frac{1}{2} \|\hat{r}(x)\|^2$

- Critère : $\hat{f}(x) = \frac{1}{2} \|\hat{r}(x)\|^2 = \frac{1}{2} \hat{r}(x)^T \hat{r}(x)$

$$\Rightarrow \hat{f}(x) = \frac{1}{2} \left(r(x_k) + \nabla r(x_k)^T (x - x_k) \right)^T \left(r(x_k) + \nabla r(x_k)^T (x - x_k) \right)$$

$$= \frac{1}{2} r(x_k)^T r(x_k) + (x - x_k)^T \nabla r(x_k) r(x_k) + \frac{1}{2} (x - x_k)^T \nabla r(x_k) \nabla r(x_k)^T (x - x_k)$$
- Gradient : $\nabla \hat{f}(x) = \nabla r(x_k) \nabla r(x_k)^T (x - x_k) + \nabla r(x_k) r(x_k)$
- Condition d'ordre 1 : $\nabla \hat{f}(x) = 0 \Rightarrow \nabla r(x_k) \nabla r(x_k)^T (x - x_k) = -\nabla r(x_k) r(x_k)$

→ On retrouve l'**itération de Gauss-Newton**
 obtenue en ignorant les dérivées secondes dans le hessien

2.5.3 Moindres carrés linéaires

Cas linéaire

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|r(x)\|^2 \quad \text{avec } r(x) = Ax - b, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \quad \rightarrow \text{problème (MC)}$$

- Critère : $f(x) = \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} (Ax - b)^T (Ax - b) = \frac{1}{2} x^T A^T A x - b^T A x + \frac{1}{2} b^T b$
- Gradient : $\nabla f(x) = \nabla r(x) r(x) = A^T (Ax - b)$
- Hessien : $\nabla^2 f(x) = A^T A$

Dans le cas linéaire, la méthode de Gauss-Newton est identique à la méthode de Newton.
→ La convergence est obtenue en une itération.

Equations normales

- La solution x^* du problème (MC) vérifie

$$\nabla f(x^*) = 0 \Rightarrow A^T (Ax - b) = 0 \Rightarrow A^T A x = A^T b$$

→ système d'**équations normales** du problème (MC) : $\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|^2$

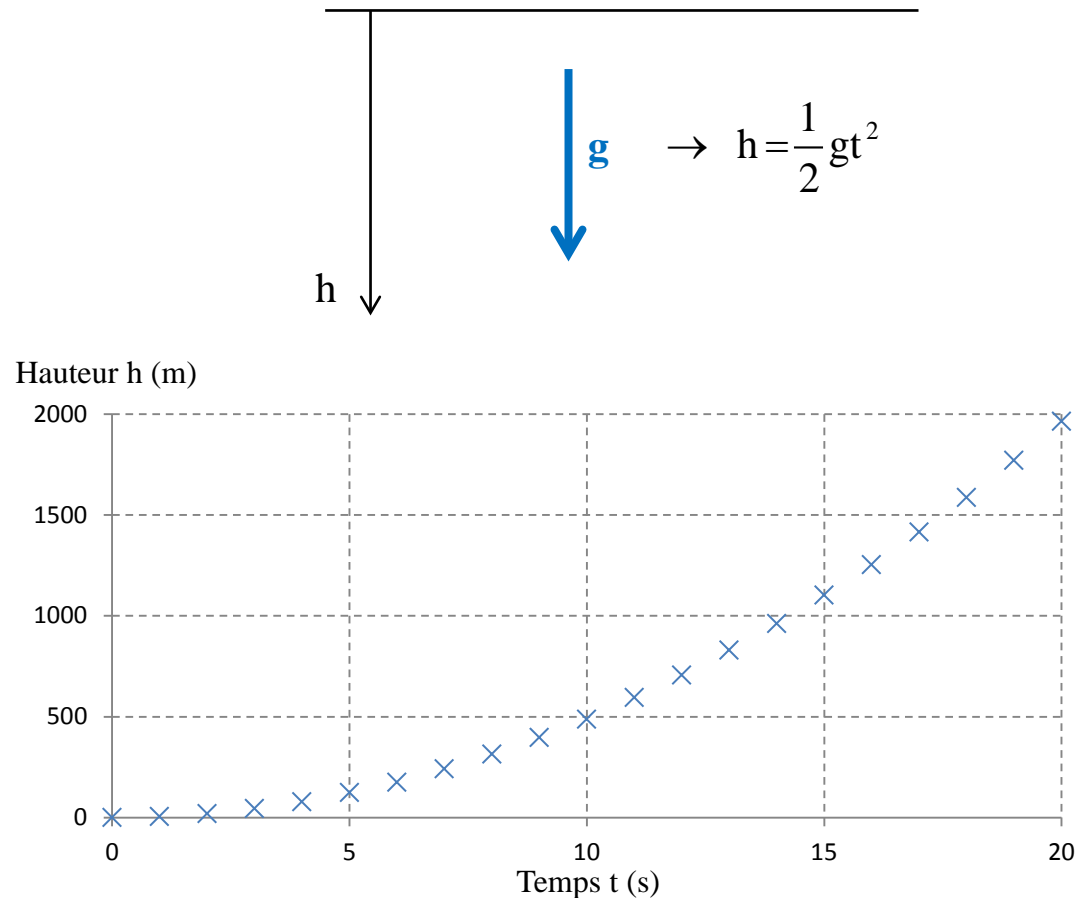
- Si A est de rang plein, x^* est l'unique solution du problème (MC).

2.5.3 Exemple

Moindres carrés linéaires

On cherche à estimer l'accélération de la gravité à partir de mesures de hauteur en chute libre.

Temps (s)	Hauteur (m)
0	0,90
1	5,40
2	20,81
3	45,73
4	78,56
5	124,10
6	175,75
7	241,41
8	315,08
9	397,36
10	488,25
11	595,35
12	707,26
13	829,98
14	961,20
15	1103,14
16	1252,89
17	1415,55
18	1586,62
19	1770,20
20	1964,29



2.5.3 Exemple

Moindres carrés linéaires

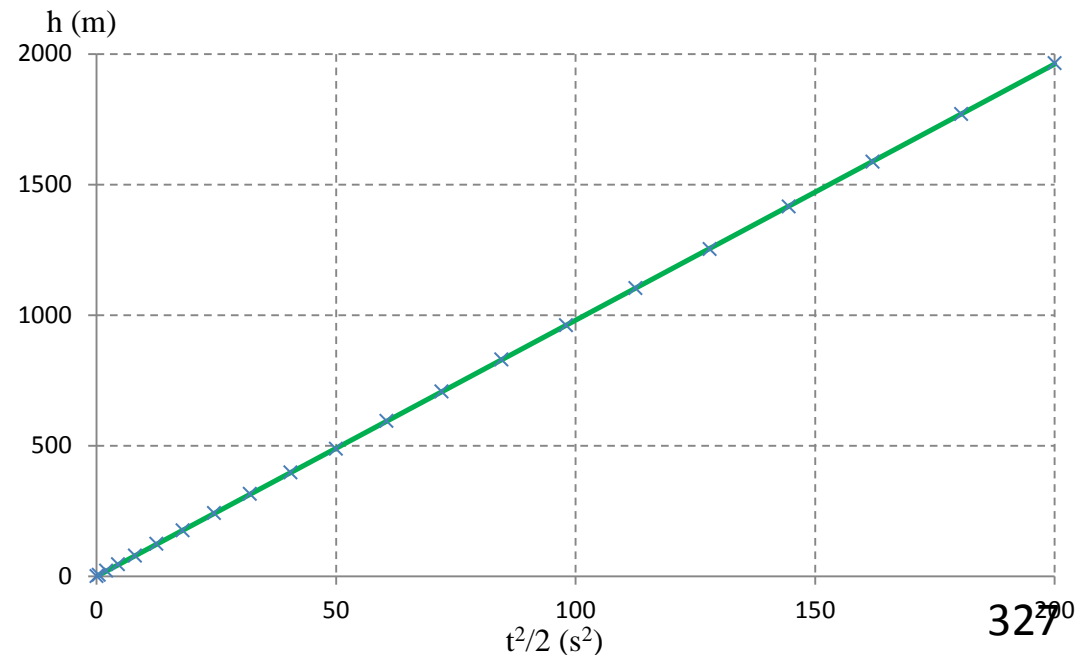
On résout le problème de moindres carrés linéaires : $\min_{g \in \mathbb{R}} f(g) = \frac{1}{2} \sum_{i=1}^m \left(h_i - g \frac{t_i^2}{2} \right)^2$

Modèle : $h_{\text{modèle}} = g \frac{t_i^2}{2}$

Résidu : $r = h_{\text{mesure}} - h_{\text{modèle}} \rightarrow f(g) = \frac{1}{2} \sum_{i=1}^m r_i(g)^2$

Solution moindres carrés : $g^* = 9,8070 \text{ m/s}^2 \rightarrow f(g^*) = 21,668$

Temps (s)	H mesure (m)	H modèle (m)	Résidu (m)
0	0,90	0,00	0,90
1	5,40	4,90	0,49
2	20,81	19,61	1,19
3	45,73	44,13	1,60
4	78,56	78,46	0,10
5	124,10	122,59	1,51
6	175,75	176,53	-0,78
7	241,41	240,27	1,13
8	315,08	313,82	1,25
9	397,36	397,18	0,17
10	488,25	490,35	-2,10
11	595,35	593,32	2,02
12	707,26	706,10	1,15
13	829,98	828,69	1,28
14	961,20	961,09	0,12
15	1103,14	1103,29	-0,14
16	1252,89	1255,30	-2,40
17	1415,55	1417,11	-1,56
18	1586,62	1588,73	-2,11
19	1770,20	1770,16	0,04
20	1964,29	1961,40	2,89



2.5.4 Filtre de Kalman

Moindres carrés récursifs

- On considère un problème de moindres carrés linéaires composé de 2 blocs

$$\min_{x \in \mathbb{R}^n} \|A_1 x - b_1\|^2 + \|A_2 x - b_2\|^2 \quad \text{avec} \quad \begin{cases} A_1 \in \mathbb{R}^{m_1 \times n}, b_1 \in \mathbb{R}^{m_1} \\ A_2 \in \mathbb{R}^{m_2 \times n}, b_2 \in \mathbb{R}^{m_2} \end{cases}$$

- Les 2 blocs correspondent à 2 séries de mesures donnant chacune des résidus respectifs.

$$\begin{cases} r_1(x) = A_1 x - b_1 & \rightarrow m_1 \text{ mesures} \\ r_2(x) = A_2 x - b_2 & \rightarrow m_2 \text{ mesures} \end{cases}$$

- Les matrices A_1 et A_2 sont supposées de rang plein $\rightarrow A_1^T A_1$ et $A_2^T A_2$ inversibles

Problème initial

- On appelle **problème initial** le problème restreint au 1^{er} bloc : $\min_{x \in \mathbb{R}^n} \|A_1 x - b_1\|^2$
- On note x_1 la solution du problème initial.
 La solution x_1 vérifie les équations normales du problème initial : $A_1^T A_1 x_1 = A_1^T b_1$
- On cherche ensuite à exprimer la solution x_2 du **problème complet** en fonction de x_1 .

2.5.4 Filtre de Kalman

Problème complet

- On réécrit le problème complet en regroupant les 2 blocs.

$$\min_{x \in \mathbb{R}^n} \|A_1 x - b_1\|^2 + \|A_2 x - b_2\|^2 \Leftrightarrow \min_{x \in \mathbb{R}^n} \left\| \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} x - \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right\|^2 \Leftrightarrow \min_{x \in \mathbb{R}^n} \|Ax - b\|^2$$

$$\text{avec } A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \in \mathbb{R}^{(m_1+m_2) \times n}, b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \in \mathbb{R}^{m_1+m_2}$$

- La solution x_2 vérifie les **équations normales du problème complet**.

$$A^T A x_2 = A^T b \Rightarrow \begin{pmatrix} A_1^T & A_2^T \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} x_2 = \begin{pmatrix} A_1^T & A_2^T \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \Rightarrow (A_1^T A_1 + A_2^T A_2) x_2 = A_1^T b_1 + A_2^T b_2$$

- La solution x_1 vérifie les **équations normales du problème initial** : $A_1^T A_1 x_1 = A_1^T b_1$

- On exprime x_2 en fonction de x_1 et des données du 2^{ème} bloc de mesures

$$\begin{aligned} (A_1^T A_1 + A_2^T A_2) x_2 &= A_1^T b_1 + A_2^T b_2 = A_1^T A_1 x_1 + A_2^T b_2 = A_1^T A_1 x_1 + A_2^T A_2 x_1 - A_2^T A_2 x_1 + A_2^T b_2 \\ &= (A_1^T A_1 + A_2^T A_2) x_1 + A_2^T (b_2 - A_2 x_1) \end{aligned}$$

$$\Rightarrow x_2 = x_1 + (A_1^T A_1 + A_2^T A_2)^{-1} A_2^T (b_2 - A_2 x_1)$$

2.5.4 Filtre de Kalman

Filtre de Kalman

La solution des moindres carrés est mise à jour à chaque nouvelle mesure disponible.
 → traitement incrémental en temps réel

- Initialisation :
 filtre H_0 (=0 par défaut)
 solution x_0 (=0 par défaut)
 - Itération :
 filtre H_k $H_k = H_{k-1} + A_k^T A_k$
 solution x_k $x_k = x_{k-1} + H_k^{-1} A_k^T (b_k - A_k x_{k-1})$
- $$\Leftrightarrow \min_{x \in \mathbb{R}^n} \sum_{i=1}^k \|A_i x - b_i\|^2$$

Mise en œuvre

- Les matrices $A_k^T A_k$ doivent être inversibles → en prenant un nombre suffisant de mesures
- On peut introduire une **pondération** ρ pour privilégier les mesures récentes.
 → $\rho = 1$ pour donner le même poids à toutes les mesures
 → $\rho < 1$ pour privilégier la dernière mesure

- Itération :
 filtre H_k $H_k = \rho H_{k-1} + A_k^T A_k$
 solution x_k $x_k = x_{k-1} + H_k^{-1} A_k^T (b_k - A_k x_{k-1})$
- $$\Leftrightarrow \min_{x \in \mathbb{R}^n} \sum_{i=1}^k \rho^{k-i} \|A_i x - b_i\|^2$$

2.5.4 Filtre de Kalman

Exemple

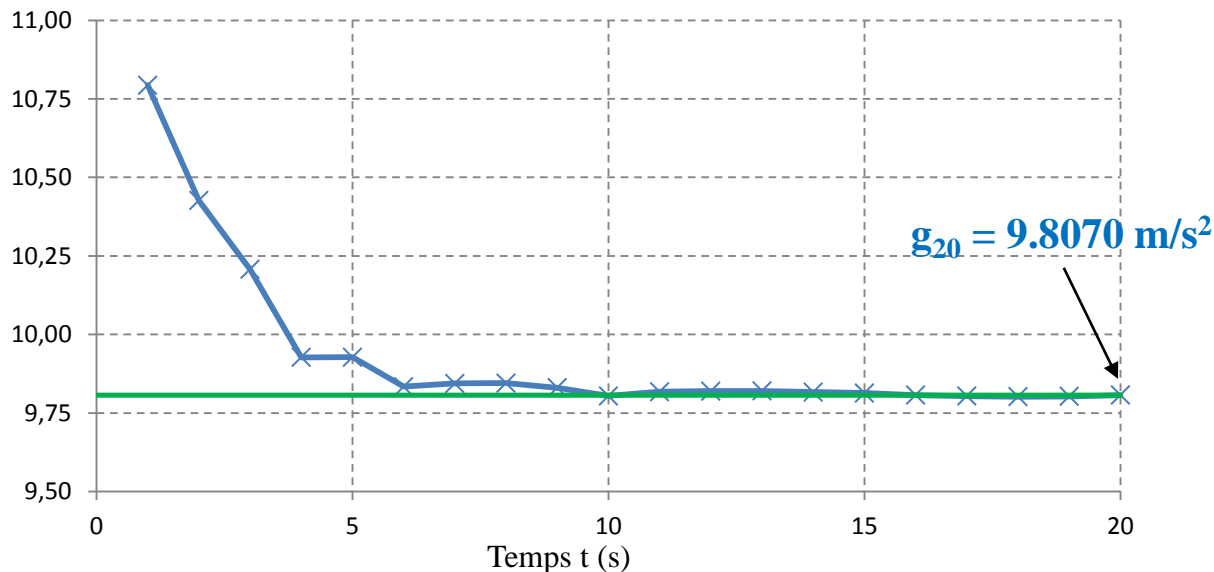
On applique la mise à jour de Kalman à chaque nouvelle mesure.

Mesure : $h_{\text{mesure}} = b \rightarrow b_k = h_k$

Modèle : $h_{\text{modèle}} = Ax = \frac{t^2}{2}g \rightarrow A_k = \frac{t_k^2}{2}, x_k = g_k$

- Filtre : $\begin{cases} H_0 = 0 \\ x_0 = 0 \end{cases} \rightarrow \begin{cases} H_k = H_{k-1} + A_k^T A_k \\ x_k = x_{k-1} + H_k^{-1} A_k^T (b_k - A_k x_{k-1}) \end{cases}$

Estimation g (m/s²)



Temps (s)	Matrice H	Estimation g (m/s²)
0	0,0	0,0000
1	0,3	10,7937
2	4,3	10,4263
3	24,5	10,2074
4	88,5	9,9269
5	244,8	9,9274
6	568,8	9,8341
7	1169,0	9,8440
8	2193,0	9,8450
9	3833,3	9,8305
10	6333,3	9,8046
11	9993,5	9,8177
12	15177,5	9,8195
13	22317,8	9,8204
14	31921,8	9,8167
15	44578,0	9,8136
16	60962,0	9,8068
17	81842,3	9,8041
18	108086,3	9,8016
19	140666,5	9,8029
20	180666,5	9,8070

2.5.4 Filtre de Kalman

Comparaison moindres carrés – filtre de Kalman

- Les moindres carrés sont adaptés à l'estimation de paramètres de modèle **a posteriori** (= lorsque toutes les mesures sont disponibles).
- Le filtre de Kalman est adapté à des applications **en temps réel** (= lorsque les mesures arrivent une à une)
- Les moindres carrés et le filtre de Kalman sont équivalents pour l'estimation de paramètres d'un modèle linéaire.
Le filtre de Kalman peut être appliqué à différents problèmes d'estimation.

Applications du filtre de Kalman

- Estimation de l'état d'un système dynamique linéaire, discret ou continu.
→ état fonction du temps, mesures en temps réel
- Estimation de l'état d'un système dynamique non linéaire
→ par linéarisation autour d'une trajectoire de référence estimée au préalable
- Méthode de filtrage largement utilisée dans les systèmes temps réels
→ nombreuses variantes

2.5.5 Gradient conjugué

Problème quadratique

On considère un problème de **moindres carrés linéaires**.

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|r(x)\|^2 \quad \text{avec } r(x) = Ax - b, \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$
$$\Rightarrow \frac{1}{2} \|r(x)\|^2 = \frac{1}{2} (Ax - b)^T (Ax - b) = \frac{1}{2} x^T A^T A x - b^T A x + \frac{1}{2} b^T b$$

Le problème de moindres carrés revient à minimiser une **forme quadratique définie positive**.

$$f(x) = \frac{1}{2} x^T Q x + c^T x \quad \text{avec } Q \in \mathbb{R}^{n \times n} \text{ symétrique définie positive}$$

Méthode de directions conjuguées

- 2 directions d_i et d_j sont **conjuguées** par rapport à Q si $d_i^T Q d_j = 0$
- On obtient le minimum de la forme quadratique définie positive en n itérations à pas optimal suivant des directions conjuguées $(d_i)_{i=1, \dots, n}$.
- La **méthode du gradient conjugué** consiste à construire une suite de directions conjuguées et à minimiser suivant ces directions successives pour obtenir le minimum de f .
→ extension à une fonction f quelconque (méthode de Fletcher-Reeves)

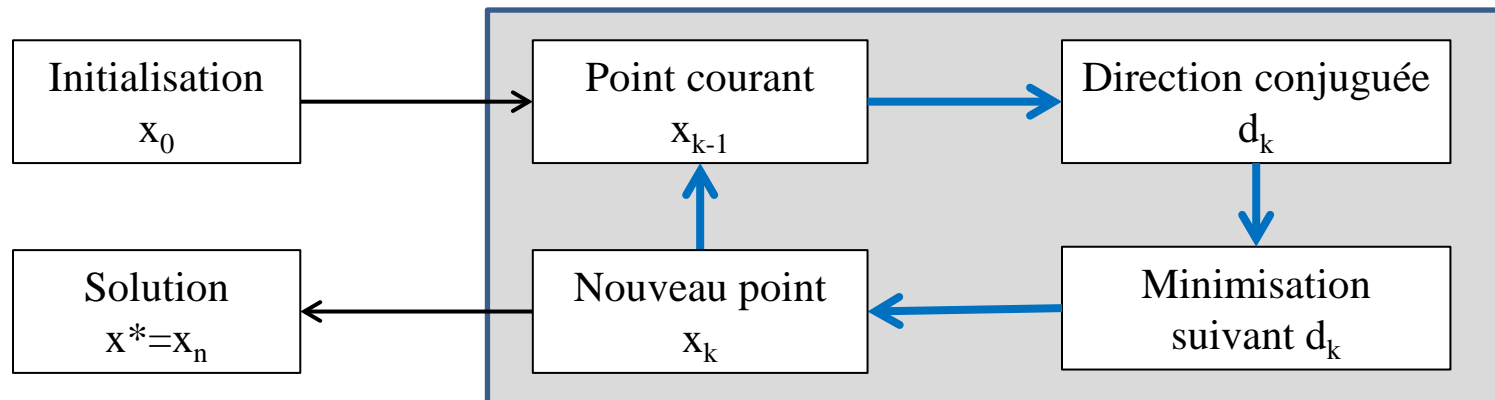
2.5.5 Gradient conjugué

Méthode du gradient conjugué

- On part d'un point initial x_0 quelconque. La direction initiale est le gradient : $d_1 = -\nabla f(x_0)$
- A chaque itération, le nouveau point x_k est obtenu par minimisation suivant d_k .

$$\min_s f(x_{k-1} + s d_k) \rightarrow x_k = x_{k-1} + s_k d_k$$

- La nouvelle direction d_{k+1} est définie à partir de la précédente d_k et du gradient en x_k .
 $d_{k+1} = -\nabla f(x_k) + \beta_k d_k \rightarrow$ plusieurs méthodes possibles
- La méthode converge en n itérations pour une fonction quadratique définie positive. Pour une fonction f quelconque, la convergence est généralement rapide car f est proche d'une fonction quadratique définie positive au voisinage de l'optimum.



2.5.5 Gradient conjugué

Méthode du gradient conjugué

- La direction de descente à l'itération k est cherchée sous la forme $\mathbf{d}_{k+1} = -\nabla f(\mathbf{x}_k) + \beta_k \mathbf{d}_k$ avec \mathbf{x}_k obtenu par minimisation suivant \mathbf{d}_k : $\min_s f(\mathbf{x}_{k-1} + s\mathbf{d}_k) \rightarrow \mathbf{x}_k = \mathbf{x}_{k-1} + s_k \mathbf{d}_k$
- La nouvelle direction \mathbf{d}_{k+1} doit être conjuguée à toutes les directions précédentes $\mathbf{d}_1, \dots, \mathbf{d}_k$.
 $\mathbf{d}_{k+1}^T \mathbf{Q} \mathbf{d}_i = 0, i = 1, \dots, k$

Il existe plusieurs méthodes de construction des directions conjuguées successives $(\mathbf{d}_i)_{i=1, \dots, n}$.

- Directions de **Fletcher-Reeves**

$$\mathbf{d}_{k+1} = -\nabla f(\mathbf{x}_k) + \beta_k \mathbf{d}_k \quad \text{avec} \quad \beta_k = \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\|\nabla f(\mathbf{x}_{k-1})\|^2}$$

- Directions de **Polak-Ribière**

$$\mathbf{d}_{k+1} = -\nabla f(\mathbf{x}_k) + \beta_k \mathbf{d}_k \quad \text{avec} \quad \beta_k = \frac{(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}))^T \nabla f(\mathbf{x}_k)}{\|\nabla f(\mathbf{x}_{k-1})\|^2}$$

→ formules équivalentes dans le cas d'une fonction f quadratique

2.5.5 Gradient conjugué

Directions de Fletcher-Reeves

$$d_{k+1} = -\nabla f(x_k) + \beta_k d_k \quad \text{avec} \quad \beta_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2} \quad \rightarrow \text{directions } \textbf{conjuguées} \text{ pour}$$

$$f(x) = \frac{1}{2} x^T Q x + c^T x$$

Preuve : On note $g_k = \nabla f(x_k) = Qx_k + c$

- **Propriété préliminaire**

*Si les n directions d_1, \dots, d_n sont conjuguées, alors $d_i^T g_k = 0$, $i = 1, \dots, k$ pour $k=1$ à n .
 et $g_i^T g_k = 0$, $i = 1, \dots, k-1$*

A l'itération i , x_i est obtenu par minimisation suivant d_i

$$\min_s f(x_{i-1} + s d_i) \Rightarrow \frac{d}{ds} f(x_{i-1} + s d_i) = d_i^T \nabla f(x_{i-1} + s d_i) = 0 \Rightarrow d_i^T g_i = 0, i = 1, \dots, n$$

$$x_k = x_i + \sum_{j=i+1}^k s_j d_j \Rightarrow g_k = Qx_k + c = Q \left(x_i + \sum_{j=i+1}^k s_j d_j \right) + c = \nabla f(x_i) + \sum_{j=i+1}^k s_j Q d_j$$

$$\Rightarrow d_i^T g_k = d_i^T g_i + \sum_{j=i+1}^k s_j d_i^T Q d_j = 0 \quad \text{car} \quad \begin{cases} d_i^T g_i = 0 \\ d_i^T Q d_j = 0 \text{ si } i \neq j \end{cases}$$

$$\Rightarrow g_i^T g_k = g_i^T g_k - \beta_i d_i^T g_k \quad \text{car } d_i^T g_k = 0, i = 1, \dots, k$$

$$= -d_{i+1}^T g_k = 0 \quad \text{car } i < k$$

2.5.5 Gradient conjugué

Directions de Fletcher-Reeves

Preuve : on montre par récurrence que d_{k+1} est conjuguée à d_1, \dots, d_k .

- Pour d_1 et d_2 :

$$d_2^T Q d_1 = -(g_1 + \beta_1 g_0)^T Q \frac{x_1 - x_0}{s_1} \text{ car } \begin{cases} d_1 = -g_0 \\ d_2 = -(g_1 + \beta_1 d_1) \\ x_1 = x_0 + s_1 d_1 \end{cases}$$

$$\Rightarrow d_2^T Q d_1 = -(g_1 + \beta_1 g_0)^T \frac{g_1 - g_0}{s_1} \text{ car } g = Qx + c$$

$$\Rightarrow d_2^T Q d_1 = -(\|g_1\|^2 - \beta_1 \|g_0\|^2) \text{ car } g_1^T g_0 = g_1^T d_1 = 0 \quad (\text{propriété préliminaire})$$

$$\Rightarrow d_2^T Q d_1 = 0 \text{ car } \beta_1 = \frac{\|\nabla f(x_1)\|^2}{\|\nabla f(x_0)\|^2} = \frac{\|g_1\|^2}{\|g_0\|^2} \quad (\text{par définition de } \beta)$$

- On suppose d_1, \dots, d_k conjuguées : $d_i^T Q d_j = 0$, $i \neq j$, $i, j \leq k$

Il faut montrer que $d_i^T Q d_{k+1} = 0$, $i = 1, \dots, k$

$$d_i^T Q d_{k+1} = -d_i^T Q (g_k - \beta_k d_k) = -d_i^T Q g_k \text{ car } d_i^T Q d_k = 0 \quad (\text{par hypothèse de récurrence})$$

$$\Rightarrow d_i^T Q d_{k+1} = -(Q d_i)^T g_k = -\left(Q \frac{x_i - x_{i-1}}{s_i} \right)^T g_k \text{ car } x_i = x_{i-1} + s_i d_i$$

$$\Rightarrow d_i^T Q d_{k+1} = \left(\frac{g_i - g_{i-1}}{s_i} \right)^T g_k \text{ car } g = Qx + c$$

2.5.5 Gradient conjugué

Directions de Fletcher-Reeves

Preuve

- On obtient : $d_i^T Q d_{k+1} = \frac{1}{s_i} (g_i^T g_k - g_{i-1}^T g_k) = 0$ pour $i = 1, \dots, k-1$
 car $g_i^T g_k = 0$, $i = 1, \dots, k-1$ (propriété préliminaire)
- Il faut encore montrer pour $i=k$: $d_k^T Q d_{k+1} = 0$

$$d_k^T Q d_{k+1} = 0 \Rightarrow d_k^T Q (-g_k + \beta_k d_k) = 0 \Rightarrow \beta_k = \frac{d_k^T Q g_k}{d_k^T Q d_k}$$

$$d_k^T Q g_k = \left(Q \frac{x_k - x_{k-1}}{s_k} \right)^T g_k = \frac{1}{s_k} (g_k - g_{k-1})^T g_k = \frac{1}{s_k} \|g_k\|^2$$

$$d_k^T Q d_k = d_k^T \left(Q \frac{x_k - x_{k-1}}{s_k} \right) = \frac{1}{s_k} d_k^T (g_k - g_{k-1}) = -\frac{1}{s_k} d_k^T g_{k-1} \text{ car } d_k^T g_k = 0 \text{ (propriété préliminaire)}$$

$$\Rightarrow d_k^T Q d_k = -\frac{1}{s_k} d_k^T g_{k-1} = -\frac{1}{s_k} (-g_{k-1} + \beta_{k-1} d_{k-1})^T g_{k-1} = \frac{1}{s_k} \|g_{k-1}\|^2 \text{ car } d_{k-1}^T g_{k-1}$$

$$\text{On obtient bien : } \beta_k = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \rightarrow \text{valeur de } \beta_k \text{ telle que } d_k^T Q d_{k+1} = 0$$

2.5.5 Exemple

Gradient conjugué

- Fonction quadratique $f(x_1, x_2) = \frac{1}{2}x_1^2 + x_1x_2 + x_2^2 \rightarrow \nabla f(x_1, x_2) = \begin{pmatrix} x_1 + x_2 \\ x_1 + 2x_2 \end{pmatrix}$
- Point initial $x^0 = \begin{pmatrix} 10 \\ -5 \end{pmatrix}$
- Itération 1** : $d^1 = -\nabla f(x^0) = \begin{pmatrix} -5 \\ 0 \end{pmatrix} \rightarrow x^1 = x^0 + sd^1 = \begin{pmatrix} 10-5s \\ -5 \end{pmatrix} = 5 \begin{pmatrix} 2-s \\ -1 \end{pmatrix}$
 $\min_s f(x^0 + sd^1) = \frac{25}{2}(2-s)^2 - 25(2-s) + 25 \Rightarrow s^1 = 1 \rightarrow x^1 = \begin{pmatrix} 5 \\ -5 \end{pmatrix}, \nabla f(x^1) = \begin{pmatrix} 0 \\ -5 \end{pmatrix}$
- Itération 2** : $d^2 = -\nabla f(x^1) + \beta^1 d^1$ avec $\beta^1 = \frac{\|\nabla f(x^1)\|^2}{\|\nabla f(x^0)\|^2} = \frac{25}{25} = 1$
 $d^2 = \begin{pmatrix} -5 \\ 5 \end{pmatrix} \rightarrow x^2 = x^1 + sd^2 = \begin{pmatrix} 5-5s \\ -5+5s \end{pmatrix} = 5(1-s) \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
 $\min_s f(x^1 + sd^2) = \frac{25}{2}(1-s)^2 - 25(1-s)^2 + 25(1-s)^2 \Rightarrow s^2 = 1 \rightarrow x^2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \nabla f(x^2) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- Solution : $x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \nabla f(x^*) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Sommaire

1. Bases théoriques
- 2. Optimisation sans contraintes**
 - 2.1 Méthodes de descente
 - 2.2 Méthode de Newton
 - 2.3 Recherche linéaire
 - 2.4 Région de confiance
 - 2.5 Moindres carrés
 - 2.6 Méthode de Nelder-Mead**
 - 2.6.1 Principes
 - 2.6.2 Algorithme
 - 2.6.3 Exemples
3. Optimisation avec contraintes

2.6.1 Méthode de Nelder-Mead

Problème sans contrainte

$$\min_{x \in \mathbb{R}^n} f(x)$$

Principes

La méthode de Nelder-Mead est une méthode d'**ordre 0** (sans dérivées)

→ n'utilise que des évaluations de la fonction (aucune évaluation du gradient)

- On définit un ensemble P de $n+1$ points de \mathbb{R}^n : $P = \{x_1, x_2, \dots, x_n, x_{n+1}\}$
 P est un **polytope** ou « simplexe » de \mathbb{R}^n .
- Les points de P sont rangés du meilleur au plus mauvais : $f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \leq f(x_{n+1})$
- A chaque itération, on cherche à remplacer le plus mauvais point par un point meilleur.

$$P = \{x_1, x_2, \dots, x_n, x_{n+1}\} \rightarrow P' = \{x_1, x_2, \dots, x_n, x_{\text{new}}\}$$

$$\rightarrow P' = \{x_1', x_2', \dots, x_n', x_{n+1}'\} \quad \text{après reclassement}$$
- On obtient une suite de polytopes $(P^k)_{k \in \mathbb{N}}$: $P^k = \{x_1^k, x_2^k, \dots, x_n^k, x_{n+1}^k\}$
- Si la méthode converge, les polytopes P^k sont de taille décroissante
 et les points du polytope convergent vers un minimum local de f .

2.6.1 Méthode de Nelder-Mead

Polytope

$$P = \{x_1, x_2, \dots, x_n, x_{n+1}\}$$

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \leq f(x_{n+1})$$

Nouveau point

x_c est le barycentre des **n meilleurs points**.

$$x_c = \frac{1}{n} \sum_{i=1}^n x_i$$

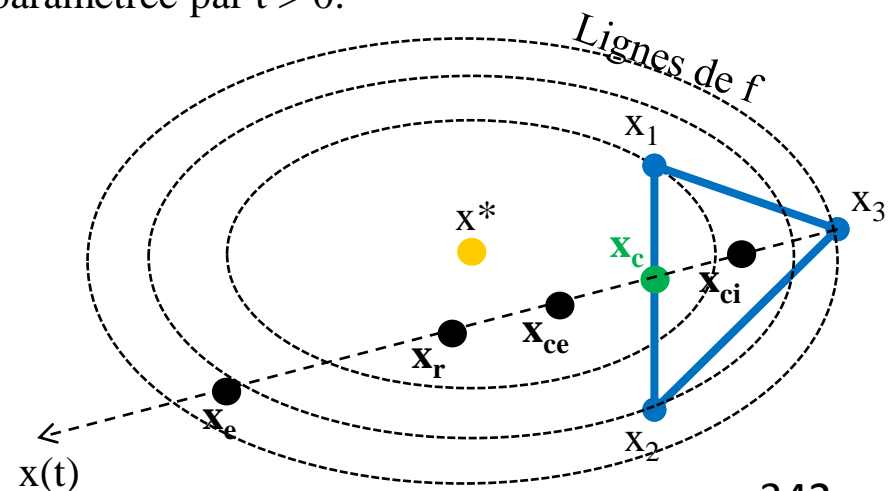
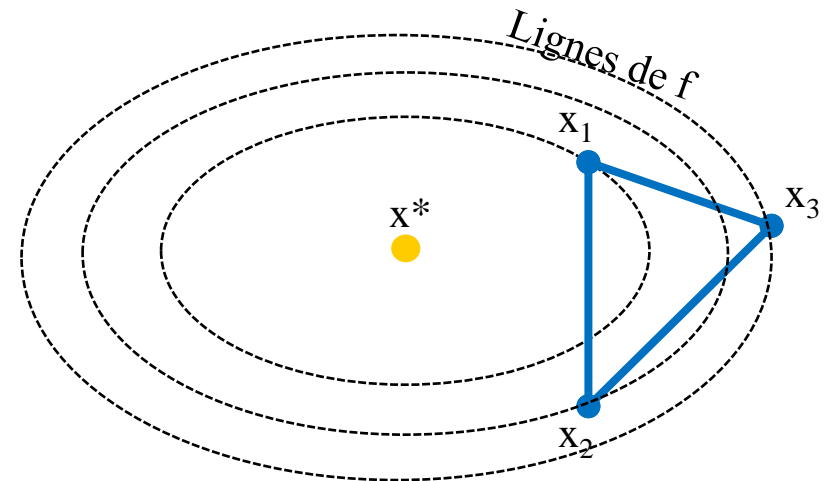
On cherche à ramener **le plus mauvais point** x_{n+1} vers le barycentre x_c .

On teste des points x sur la demi-droite $[x_{n+1}, x_c]$ paramétrée par $t > 0$.

$$x(t) = x_c + t(x_{n+1} - x_c)$$

Points testés successivement :

- **Réflexion** : $x(t=+1) = x_r$
- **Expansion** : $x(t=+2) = x_e$
- **Contraction externe** : $x(t=+0.5) = x_{ce}$
- **Contraction interne** : $x(t=-0.5) = x_{ci}$



2.6.1 Méthode de Nelder-Mead

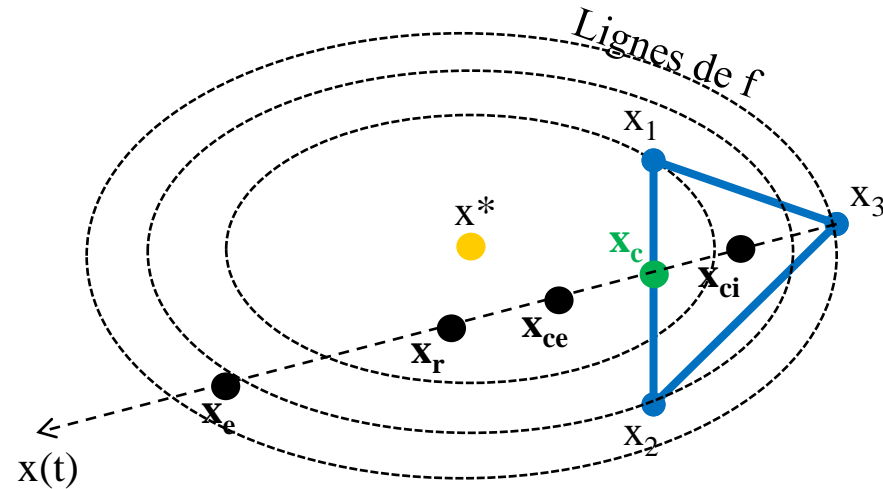
Nouveau point

Le polytope initial est $P = \{x_1, x_2, \dots, x_n, x_{n+1}\}$ avec $f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \leq f(x_{n+1})$

La demi-droite $[x_{n+1}, x_c)$ est supposée être une direction de descente.

On teste d'abord x_r (symétrique de x_{n+1} / x_c).

- Résultat très bon : $f(x_r) < f(x_1)$
 → on essaie x_e
 on remplace x_{n+1} par x_e ou x_r .
- Résultat bon : $f(x_1) \leq f(x_r) < f(x_n)$
 → on remplace x_{n+1} par x_r .
- Résultat moyen : $f(x_n) \leq f(x_r) < f(x_{n+1})$
 → on essaie x_{ce}
 on remplace x_{n+1} par x_{ce} ou x_r .
- Résultat mauvais : $f(x_{n+1}) \leq f(x_r)$
 → on essaie x_{ci}
 on remplace x_{n+1} par x_{ci} , ou on contracte tout le polytope P vers x_1



2.6.2 Algorithme

Itération k

$$P^k = \{x_1^k, x_2^k, \dots, x_n^k, x_{n+1}^k\} \quad \text{avec} \quad f(x_1^k) \leq f(x_2^k) \leq \dots \leq f(x_n^k) \leq f(x_{n+1}^k)$$

1. Barycentre :
$$x_c = \frac{1}{n} \sum_{i=1}^n x_i^k \rightarrow d = x_c - x_{n+1}^k$$

2. Réflexion :
$$x_r = x_c + d \rightarrow f(x_r)$$

3. Si $f(x_r) < f(x_1)$
 Expansion :
$$x_e = x_c + 2d \rightarrow f(x_e)$$

 Si $f(x_e) < f(x_r)$ $\rightarrow x' = x_e$
 Sinon $\rightarrow x' = x_r$

4. Si $f(x_1) < f(x_r) < f(x_n)$ $\rightarrow x' = x_r$

5. Si $f(x_n) < f(x_r) < f(x_{n+1})$
 Contraction externe :
$$x_{ce} = x_c + d/2 \rightarrow f(x_{ce})$$

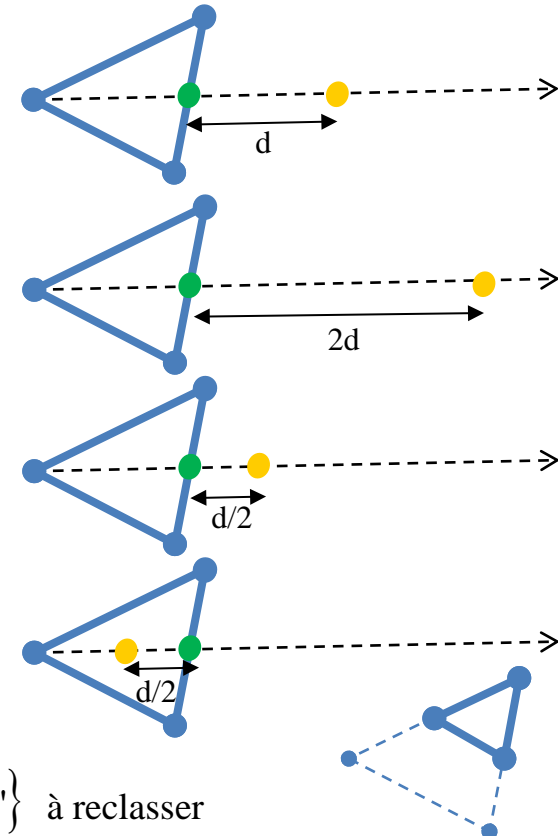
 Si $f(x_{ce}) < f(x_r)$ $\rightarrow x' = x_{ce}$
 Sinon $\rightarrow x' = x_r$

6. Si $f(x_{n+1}) < f(x_r)$
 Contraction interne :
$$x_{ci} = x_c - d/2 \rightarrow f(x_{ci})$$

 Si $f(x_{ci}) < f(x_{n+1})$ $\rightarrow x' = x_{ci}$
 Sinon réduction de P vers $x_1 \rightarrow x_i^{k+1} = \frac{1}{2}(x_1 + x_i^k)$

7. Nouveau point x' \rightarrow polytope $P^{k+1} = \{x_1^k, x_2^k, \dots, x_n^k, x'\}$ à reclasser

$$P^{k+1} = \{x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1}, x_{n+1}^{k+1}\} \quad \text{avec} \quad f(x_1^{k+1}) \leq f(x_2^{k+1}) \leq \dots \leq f(x_n^{k+1}) \leq f(x_{n+1}^{k+1})$$



2.6.2 Algorithme

Initialisation

Polytope initial $P^0 = \{x_1^0, x_2^0, \dots, x_n^0, x_{n+1}^0\}$ avec $f(x_1^0) \leq f(x_2^0) \leq \dots \leq f(x_n^0) \leq f(x_{n+1}^0)$
→ points suffisamment distants, non alignés

Itération k

$P^k = \{x_1^k, x_2^k, \dots, x_n^k, x_{n+1}^k\}$ avec $f(x_1^k) \leq f(x_2^k) \leq \dots \leq f(x_n^k) \leq f(x_{n+1}^k)$

- Nouveau point x' → reclassement des points x_1, x_2, \dots, x_n, x' par valeur de f croissante
- Nouveau polytope :

$P^{k+1} = \{x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1}, x_{n+1}^{k+1}\}$ avec $f(x_1^{k+1}) \leq f(x_2^{k+1}) \leq \dots \leq f(x_n^{k+1}) \leq f(x_{n+1}^{k+1})$

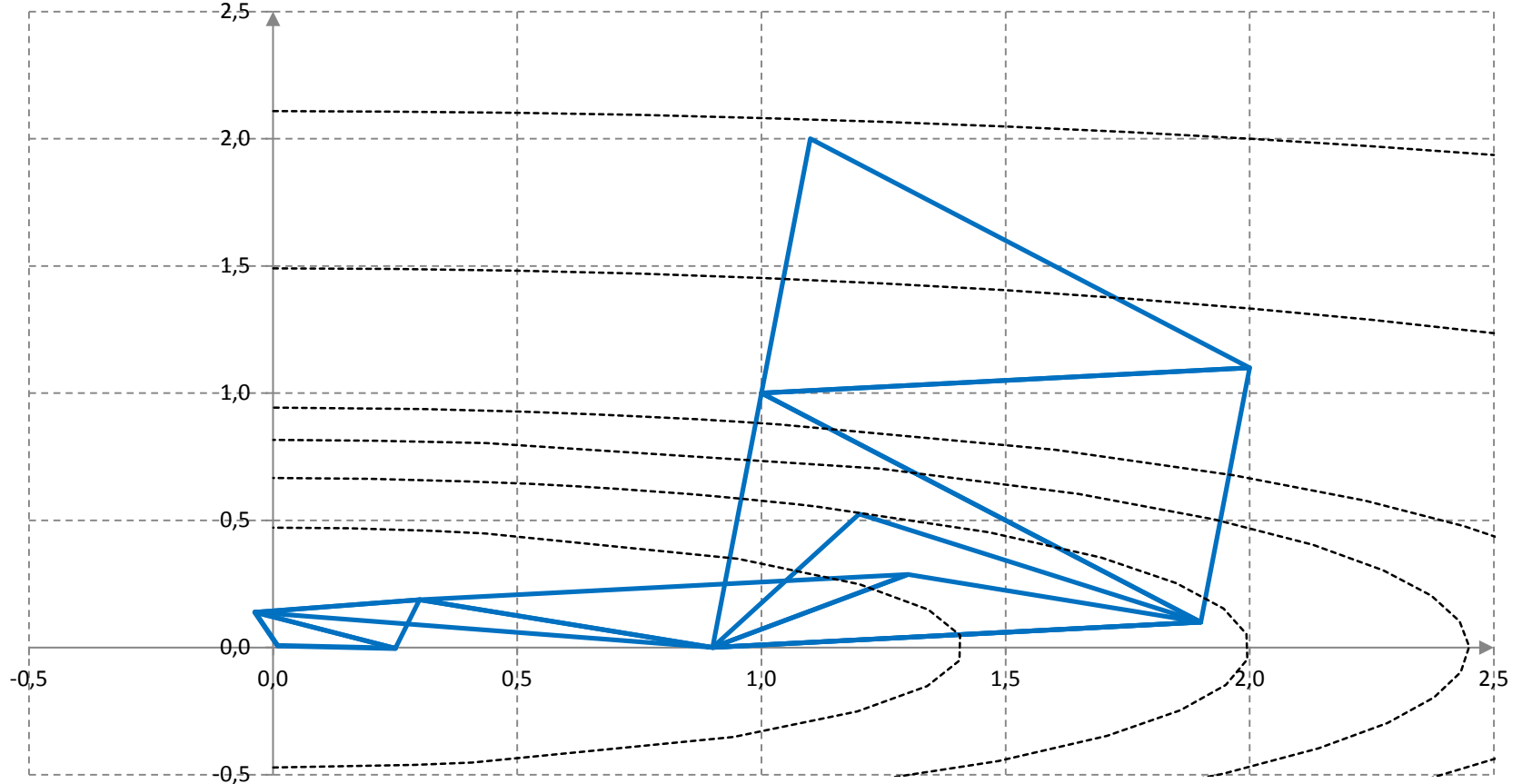
Arrêt

- Taille du polytope
- Valeur de la fonction (non décroissante, identique sur tout le polytope)
- Dégénérescence du polytope (alignement des points).

2.6.3 Exemples

Exemple 1 : fonction quadratique

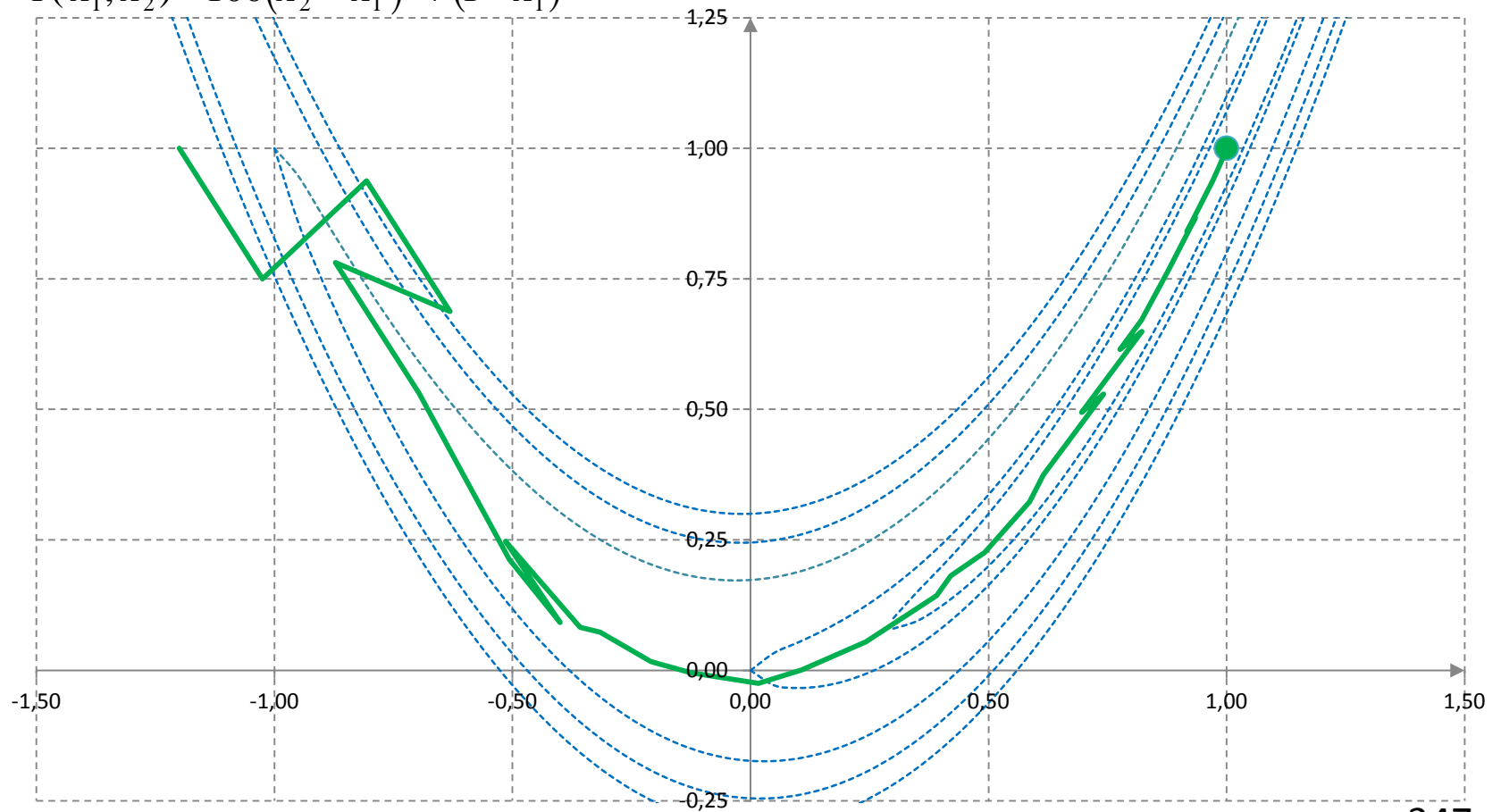
$$f(x_1, x_2) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$$



2.6.3 Exemples

Exemple 2 : fonction de Rosenbrock

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$



Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
- 3. Optimisation avec contraintes**
 - 3.1 Simplexe
 - 3.2 Point intérieur
 - 3.3 Gradient projeté
 - 3.4 Lagrangien augmenté
 - 3.5 Programmation quadratique séquentielle
 - 3.6 Convergence

3 Optimisation avec contraintes

Problème non linéaire sous contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sous} \quad \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \quad \rightarrow \text{Problème noté (PO)}$$

Catégories de problèmes

- **Programmation linéaire** \rightarrow Fonctions f, c_E, c_I linéaires
- **Programmation non linéaire** \rightarrow Fonctions f, c_E, c_I quelconques

Traitement des contraintes

- Méthodes de contraintes actives \rightarrow Identification des inégalités actives
Transformation en un **problème avec contraintes égalité**
Respect des contraintes à chaque itération
- Méthodes de point intérieur \rightarrow Fonction barrière (pénalisation intérieure)
Suivi d'un **chemin central intérieur aux contraintes**
- Méthodes de pénalisation \rightarrow Critère augmenté (pénalisation extérieure)
Transformation en un **problème sans contraintes**

3 Optimisation avec contraintes

Problème non linéaire sous contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sous} \quad \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \quad \rightarrow \text{Problème noté (PO)}$$

Classification des méthodes

	Méthode primale	Méthode primale-duale	Méthode duale
Problème traité	problème primal	problème primal	problème dual
Objectif	min f - méthode directe - point stationnaire	solution KKT - méthode indirecte - point stationnaire	max w - méthode indirecte - point col
Itérations	admissibles	admissibles ou non	non admissibles
Variables	primales x	primales x , duales λ	primales x , duales λ
Algorithmes	- simplexe (LP) - gradient projeté - pénalisation	- point intérieur (LP, NLP) - séquentiel quadratique - lagrangien augmenté	- Uzawa

Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
3. **Optimisation avec contraintes**
 - 3.1 **Simplexe**
 - 3.1.1 Problème linéaire
 - 3.1.2 Déplacement
 - 3.1.3 Initialisation
 - 3.1.4 Simplexe révisé
 - 3.1.5 Simplexe dual
 - 3.2 Point intérieur
 - 3.3 Gradient projeté
 - 3.4 Lagrangien augmenté
 - 3.5 Programmation quadratique séquentielle
 - 3.6 Convergence

3.1.1 Problème linéaire

- ☐ Forme standard
- ☐ Solution
- ☐ Recherche systématique
- ☐ Recherche optimisée
- ☐ Forme canonique

3.1.1 Problème linéaire

Problème linéaire sous forme standard

$$\min_x c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n \quad \text{rang}(A) = m \quad \rightarrow \text{problème noté (PL)}$$

Rappels

- **Base B** = $(A_{j_1}, \dots, A_{j_m})$ = m colonnes indépendantes de A \Rightarrow B inversible
 $AE = (B \ N)$ avec E = matrice de permutation de colonnes ($EE^T = I$)

- **Solution de base** :
$$x = E \begin{pmatrix} x_B \\ x_N \end{pmatrix} \begin{matrix} \rightarrow m \\ \rightarrow n - m \end{matrix} \text{ avec } \begin{cases} Ax = b \\ x_N = 0 \end{cases} \Rightarrow x = E \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$$

admissible (ou réalisable) si $x_B = B^{-1}b \geq 0$

- **Direction de base** $d_j, j \in N$

$$d_j = E \begin{pmatrix} d_{jB} \\ d_{jN} \end{pmatrix} \text{ avec } \begin{cases} d_{jB} = -B^{-1}A_j \\ E^T e_j = \begin{pmatrix} 0 \\ d_{jN} \end{pmatrix} \end{cases} \begin{matrix} \rightarrow \text{pour vérifier } Ax=b \\ \rightarrow \text{composantes nulles sauf } =1 \text{ sur la composante } j \end{matrix}$$

- **Coût réduit** = dérivée directionnelle suivant la direction de base d_j : $\bar{c}_j = c^T d_j = c_j - c_B^T B^{-1} A_j$
 Coût réduit négatif \rightarrow direction de descente

3.1.1 Solution

Problème linéaire sous forme standard

$$\min_x c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases} \quad A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n \\ \text{rang}(A) = m$$

Solution

On note P le polytope associé aux contraintes : $P = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\}$

Si le problème (PL) admet une solution, alors il existe un sommet optimal x^* .

*Preuve : On suppose que PL admet une solution de coût f^**

On considère un sommet x^ du polytope Q inclus dans P : $Q = \{x \in P / c^T x = f^*\}$*

On suppose par l'absurde que x^ n'est pas un sommet de P .*

$$x^* = \alpha^* y + (1 - \alpha^*)z \text{ avec } y, z \in P, y \neq z, 0 < \alpha^* < 1$$

$$\text{La fonction linéaire } \begin{cases} \varphi(\alpha) = \alpha c^T y + (1 - \alpha) c^T z \\ 0 \leq \alpha \leq 1 \end{cases} \text{ est minimale en } \alpha^* : \begin{cases} \varphi(\alpha^*) = c^T x^* = f^* \\ 0 < \alpha^* < 1 \end{cases}$$

$$\begin{aligned} \text{La fonction } \varphi \text{ est donc constante (sinon elle décroît d'un côté de } \alpha^*) &\Rightarrow c^T y = c^T z = f^* \\ &\Rightarrow y, z \in Q \end{aligned}$$

On a donc : $x^ = \alpha^* y + (1 - \alpha^*)z$ avec $y, z \in Q, y \neq z, 0 < \alpha^* < 1$
 en contradiction avec l'hypothèse que x est un sommet de Q .*

3.1.1 Solution

Recherche systématique

Si (PL) admet une solution x^* , x^* est un sommet du polytope P associé aux contraintes.

On peut donc trouver la solution : en parcourant tous les sommets de P (= bases)

en calculant les solutions de base associées

en conservant la meilleure solution de base réalisable

- Choix de m colonnes parmi les n colonnes de $A \rightarrow$ base B possible

- Vérification que la base est réalisable : B inversible et $B^{-1}b \geq 0$

\rightarrow Solution de base admissible :
$$x = E \begin{pmatrix} x_B \\ x_N \end{pmatrix} = E \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} \geq 0$$

- Valeur du coût associé à la base B :
$$f = c^T x = c_B^T x_B + c_N^T x_N = c_B^T x_B$$

- Sélection de la meilleure solution (f minimal)

Inconvénient

$$C_n^m = \frac{n!}{m!(n-m)!} \text{ combinaisons possibles} \rightarrow \text{inapplicable en pratique}$$

3.1.1 Exemple

Recherche systématique

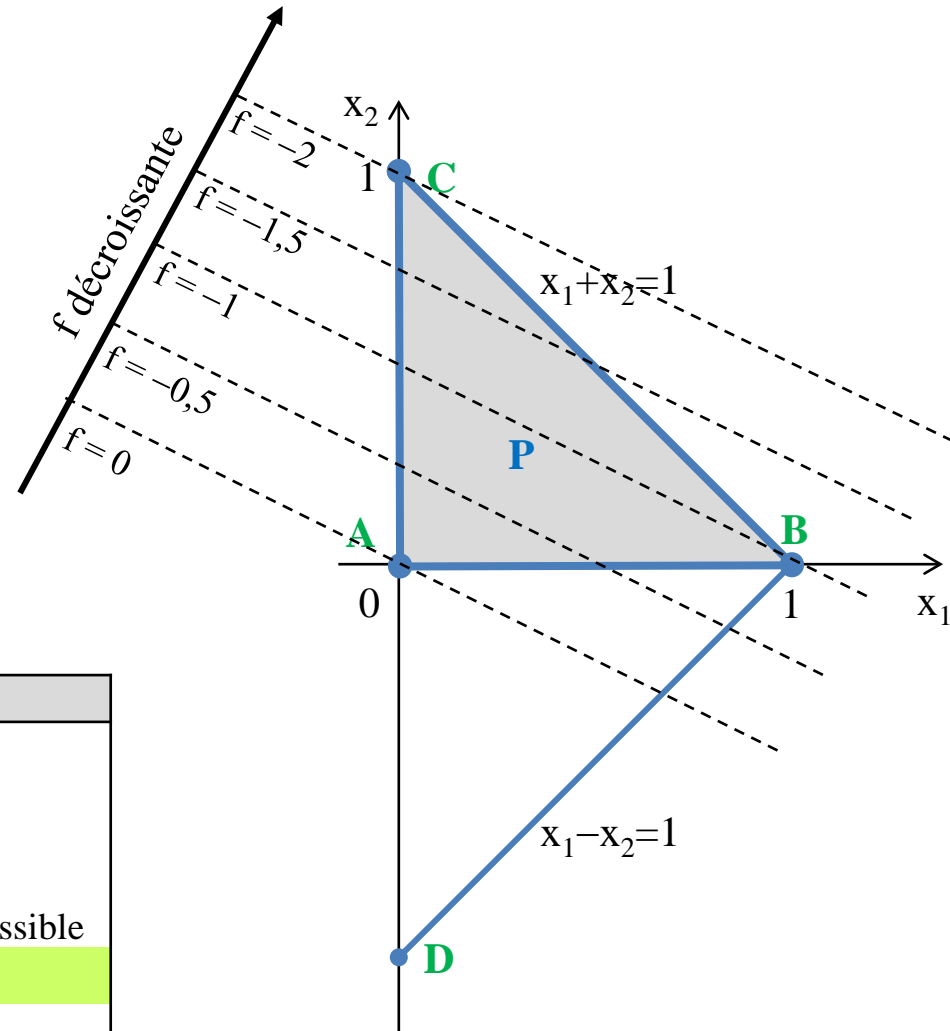
$$\min_{x_1, x_2, x_3, x_4} -x_1 - 2x_2 \text{ sous } \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 - x_2 + x_4 = 1 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

- Représentation graphique dans \mathbb{R}^2

$$P' = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} / \begin{cases} x_1 + x_2 \leq 1 \\ x_1 - x_2 \leq 1 \end{cases}, \begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \end{cases} \right\}$$

- Solution graphique : point C ($f = -2$)
- Solution par énumération des sommets

Base	x	f
x_1, x_2	$(1 \ 0 \ 0 \ 0) \rightarrow B$	-1
x_1, x_3	$(1 \ 0 \ 0 \ 0) \rightarrow B$	-1
x_1, x_4	$(1 \ 0 \ 0 \ 0) \rightarrow B$	-1
x_2, x_3	$(0 \ -1 \ 2 \ 0) \rightarrow D$	Non admissible
x_2, x_4	$(0 \ 1 \ 0 \ 2) \rightarrow C$	-2
x_3, x_4	$(0 \ 0 \ 1 \ 1) \rightarrow A$	0



3.1.1 Solution

Recherche optimisée

On évite l'énumération systématique en parcourant les sommets de façon ordonnée

→ **Méthode du simplexe** = méthode de contraintes actives

Principes

- On se déplace d'une **solution de base admissible** à une autre solution de base admissible.
→ Les solutions non admissibles ne sont pas examinées.
- Les bases successives ne diffèrent que par l'une des variables (**bases adjacentes**)
- Le déplacement d'un sommet à un autre est choisi à partir des **directions de base**
→ Déplacement suivant les arêtes du polytope
- Les **coûts réduits** déterminent les directions de descente possibles.
→ Sélection d'une direction de déplacement (plusieurs règles de sélection possibles)
- Le problème est mis sous **forme canonique** dans la base B
→ Permet de vérifier l'optimalité de la base B
→ Permet de construire le déplacement vers une base adjacente

3.1.1 Forme canonique

Réduction dans la base B

- Forme standard : $\min_x c^T x$ sous $\begin{cases} Ax = b \\ x \geq 0 \end{cases}$ $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$
 $\text{rang}(A) = m$

- Base B : $AE = (B \ N)$, $x = E \begin{pmatrix} x_B \\ x_N \end{pmatrix} \begin{matrix} \rightarrow m \\ \rightarrow n-m \end{matrix} \Rightarrow \begin{cases} Ax = Bx_B + Nx_N \\ c^T x = c_B^T x_B + c_N^T x_N \end{cases}$

- Réduction aux variables hors base

$$\begin{aligned} \min_x c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases} &\Leftrightarrow \min_{\substack{x_B \in \mathbb{R}^m \\ x_N \in \mathbb{R}^{n-m}}} c_B^T x_B + c_N^T x_N \text{ sous } \begin{cases} Bx_B + Nx_N = b \\ x_B \geq 0, x_N \geq 0 \end{cases} \\ &\Rightarrow x_B = B^{-1}b - B^{-1}Nx_N \\ &\Leftrightarrow \min_{x_N \in \mathbb{R}^{n-m}} c_B^T B^{-1}b + (c_N^T - c_B^T B^{-1}N)x_N \text{ sous } \begin{cases} x_B = B^{-1}b - B^{-1}Nx_N \geq 0 \\ x_N \geq 0 \end{cases} \end{aligned}$$

- Forme canonique dans la base B**

$$\min_{\substack{x_N \in \mathbb{R}^{n-m} \\ x_N \geq 0}} \bar{z} + \bar{c}_N^T x_N \text{ sous } x_B = \bar{b} - B^{-1}Nx_N \geq 0 \quad \text{avec } \begin{cases} \bar{b} = B^{-1}b \\ \bar{z} = c_B^T \bar{b} \\ \bar{c}_N^T = c_N^T - c_B^T B^{-1}N \end{cases}$$

→ Réduction à n-m variables = variables hors-base x_N

3.1.1 Forme canonique

Evaluation de la base B

La solution x^* du problème linéaire correspond à un sommet = solution de base admissible

→ Evaluer l'optimalité de la solution de base associée à la base B

→ Construire le déplacement vers une nouvelle base B' meilleure que B

- Forme canonique dans B :
$$\min_{\substack{x_N \in \mathbb{R}^{n-m} \\ x_N \geq 0}} \bar{z} + \bar{c}_N^T x_N \quad \text{sous} \quad x_B = \bar{b} - B^{-1} N x_N \geq 0$$
- Solution de base associée à B : $x_N = 0 \Rightarrow \begin{cases} z = \bar{z} \\ x_B = \bar{b} \geq 0 \end{cases} \quad \text{si B est admissible (ou réalisable)}$
- Variation du coût :
$$z(x_N) = \bar{z} + \bar{c}_N^T x_N = \bar{z} + \sum_{j \in N} \bar{c}_j x_j \Rightarrow \frac{\partial z}{\partial x_j} = \bar{c}_j$$

Optimalité

- Coût réduit \bar{c}_j = dérivée directionnelle suivant la direction de base d_j associée à x_j , $j \in N$
- Si tous les coûts réduits sont positifs ou nuls, la solution est optimale.
- Sinon le coût décroît suivant une direction de base d_j de coût réduit négatif
 = direction de descente

3.1.2 Déplacement

- ☐ Règles de déplacement
- ☐ Changement de base
- ☐ Formules de pivotage
- ☐ Méthode des tableaux

3.1.2 Règles de déplacement

Notations

- Matrices : B, N = matrice de base et hors base $AE = (B \ N)$
- Par extension : B, N = numéros des variables de base x_B et hors base x_N
- Solution de base associée à B : $E^T x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ avec $\begin{cases} Ax = b \\ x_N = 0 \end{cases} \Rightarrow Bx_B = b$

Direction de déplacement

- Si tous les coûts réduits sont positifs ou nuls, la solution courante est optimale.
- Sinon on choisit un indice hors base $e \in N$ de coût réduit strictement négatif : $\boxed{\bar{c}_e < 0}$
 → La direction de base associée d_e est une direction de descente.
- On se déplace à partir de x d'un pas $\alpha \geq 0$ suivant la direction de base $d_e \rightarrow x' = x + \alpha d_e$
- Le nouveau point $x' = x + \alpha d_e$ doit rester admissible :

$$\begin{cases} Ax' = b \\ x' \geq 0 \end{cases} \Leftrightarrow \begin{cases} A(x + \alpha d_e) = b \\ x + \alpha d_e \geq 0 \end{cases} \Rightarrow \begin{cases} \alpha A d_e = 0 & \text{car } Ax = b \rightarrow \text{vérifié par définition de } d_e \\ x_N + \alpha d_{eN} \geq 0 & \rightarrow \text{vérifié car } x_N = 0 \text{ et } d_{eN} \geq 0 \\ x_B + \alpha d_{eB} \geq 0 & \rightarrow \text{limite le déplacement} \end{cases}$$

- Le déplacement est limité par le fait que les variables de base doivent rester positives.

3.1.2 Règles de déplacement

Pas de déplacement

Le déplacement α suivant la direction de base d_e est limité par les contraintes $x_B \geq 0$

$$x_B + \alpha d_{eB} \geq 0 \rightarrow \text{Borne } \alpha_i \text{ pour chacune des } m \text{ variables de base } x_i, i \in B$$

- Si la composante d_{eB_i} est positive, le pas n'est pas borné : $\alpha_i = +\infty$
- Si la composante d_{eB_i} est négative, le pas est borné par : $\alpha_i = \frac{x_i}{-d_{eB_i}}$ (annulation de x_i)

Déplacement maximal

On note s le numéro de la 1^{ère} variable de base x_i qui s'annule suivant la direction d_e .

→ Le **pas maximal admissible** suivant la direction d_e est : $\alpha_s = \min_{i \in B} \alpha_i$

- Si $d_{eB} \geq 0$, le pas n'est pas borné suivant d_e
 → Le problème PL n'a pas de solution (problème non borné).
- Sinon on réalise le pas maximal α_s suivant la direction d_e : $x' = x + \alpha_s d_e$
 → **Changement de base ou pivotage**
 → Echange des variables x_s (variable de base sortant de la base courante)
 et x_e (variable hors base entrant dans la nouvelle base)

3.1.2 Changement de base

Pivotage

La direction de base d_e associé à la variable hors base x_e est définie par :

$$E^T d_e = \begin{pmatrix} d_{eB} \\ d_{eN} \end{pmatrix} \text{ avec } \begin{cases} d_{eB} = -B^{-1}A_e & \rightarrow \text{pour vérifier } Ax=b \\ E^T e_e = \begin{pmatrix} 0 \\ d_{eN} \end{pmatrix} & \rightarrow \text{composantes nulles sauf } =1 \text{ sur la composante } e \end{cases}$$

- Le nouveau point $x' = x + \alpha_s d_e$ est admissible car : - la direction d_e est admissible
 - le pas α_s respecte les contraintes $x' \geq 0$
- Variables de base $x_i, i \in B$: $x_i' = x_i + \alpha_s (d_{eB})_i$ $\begin{cases} \geq 0 & \text{si } i \neq s \\ = 0 & \text{si } i = s \end{cases} \rightarrow \begin{cases} \text{car } \alpha_s \leq \alpha \\ \text{par construction du pas } \alpha_s \end{cases}$
- Variables hors base $x_j, j \in N$: $x_j' = x_j + \alpha_s (d_{eN})_j$ $\begin{cases} = 0 & \text{si } j \neq e \\ = \alpha_s \geq 0 & \text{si } j = e \end{cases} \text{ car } (d_{eN})_j \begin{cases} = 0 & \text{si } j \neq e \\ = 1 & \text{si } j = e \end{cases}$

Nouvelle base

- Nouvelles variables hors base : $\begin{cases} x_j' = 0 & \text{pour } j \in N - \{e\} \\ x_s' = 0 & \text{pour } i = s \end{cases} \Rightarrow N' = N - \{e\} + \{s\}$
- Nouvelles variables de base : $\begin{cases} x_i' \geq 0 & \text{pour } i \in B - \{s\} \\ x_e' \geq 0 & \text{pour } j = e \end{cases} \Rightarrow B' = B - \{s\} + \{e\}$

3.1.2 Changement de base

Variation du coût

- Le nouveau coût est : $c^T x' = c^T (x + \alpha_s d_e) = c^T x + \alpha_s \bar{c}_e \Rightarrow z' = z + \alpha_s \bar{c}_e \leq z$
- Si la base n'est pas dégénérée ($x_B > 0$), toutes les directions de base sont admissibles
 - Déplacement non nul possible : $\alpha_s > 0$
 - Le coût décroît strictement : $z' < z$ car on a choisi $e \in N$ tel que $\bar{c}_e < 0$

Méthode pratique

- La nouvelle base ne diffère de la base courante que par une seule variable (= une colonne de A)
 - Limitations des calculs correspondant à un pivotage
 - **Méthode des tableaux**
- Les variables hors base sont constantes ou croissantes suivant les directions de base.
 - Toutes les variables hors base sont candidates pour entrer dans la base.
- Plusieurs règles de choix sont possibles pour la variable entrant dans la base.
 - **Règles de pivotage**
- L'algorithme nécessite une base initiale admissible.
 - Etape préliminaire de détermination de la **base initiale**

3.1.2 Changement de base

Règles de pivotage

- Choix de la variable entrante → différents choix possibles
- Détermination de la variable sortante → imposé

Variable entrante

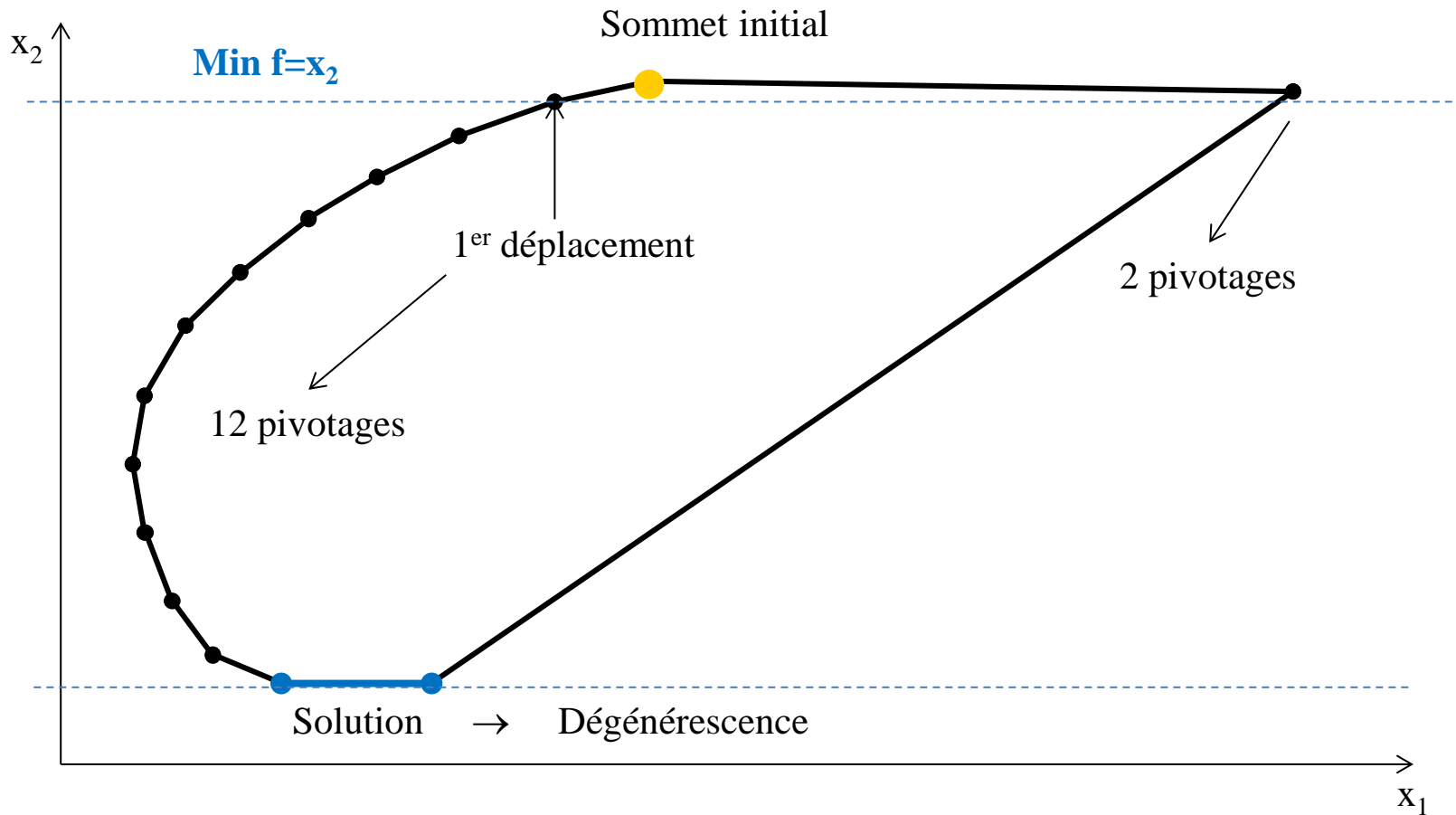
- La variable hors base entrant dans la base doit avoir un **coût réduit négatif**.
- Choix de la variable de plus petit indice
→ **Règle de Bland** (évite le cyclage pouvant se produire lorsque une base est dégénérée)
- Choix de la variable de coût réduit le plus négatif (plus forte descente)
→ **1^{ère} règle de Dantzig**
- Choix de la variable conduisant à la plus forte diminution de la fonction coût
- Choix aléatoire avec une probabilité proportionnelle au coût réduit

Variable sortante

- La variable de base sortant de la base est la 1^{ère} à s'annuler suivant la direction de base choisie
→ **2^{ème} règle de Dantzig**

3.1.2 Changement de base

Illustration



3.1.2 Formules de pivotage

Forme canonique

On écrit le problème sous forme canonique dans la base B.

- Formulation matricielle

$$\min_{\substack{x_N \in \mathbb{R}^{n-m} \\ x_N \geq 0}} \bar{z} + \bar{c}_N^T x_N \quad \text{sous} \quad x_B = \bar{b} - B^{-1} N x_N \geq 0 \quad \text{avec} \quad \begin{cases} \bar{b} = B^{-1} b \\ \bar{z} = c_B^T \bar{b} \\ \bar{c}_N^T = c_N^T - c_B^T B^{-1} N \end{cases}$$

- Formulation explicite en fonction des variables hors base $x_j, j \in N$

$$\min_{x_N \geq 0} \bar{z} + \sum_{j \in N} \bar{c}_j x_j \quad \text{sous} \quad x_i = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} x_j \geq 0, i \in B \quad \text{avec} \quad \begin{cases} \bar{b} = B^{-1} b \\ B^{-1} N = (\bar{a}_{ij})_{i \in B, j \in N} \\ \bar{z} = \sum_{i \in B} c_i \bar{b}_i \\ \bar{c}_j = c_j - \sum_{i \in B} c_i \bar{a}_{ij}, j \in N \end{cases}$$

3.1.2 Formules de pivotage

Changement de base

- Le pivotage consiste à remplacer la variable hors base x_e (entrante) : $B' = B - \{s\} + \{e\}$
 par la variable de base x_s (sortante) : $N' = N - \{e\} + \{s\}$

- Forme canonique dans la **base B** : $z = \bar{z} + \sum_{j \in N} \bar{c}_j x_j$ $x_i = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} x_j$, $i \in B$

→ Expression en fonction des variables x_j , $j \in N$

- Forme canonique dans la **base B'** : $z = \bar{z}' + \sum_{j \in N'} \bar{c}_j' x_j$ $x_i = \bar{b}_i' - \sum_{j \in N'} \bar{a}_{ij}' x_j$, $i \in B'$

→ Expression en fonction des variables x_j , $j \in N' = N - \{e\} + \{s\}$

Pour passer de la forme canonique dans la base B à la forme canonique dans la base B', il faut :

- exprimer x_e en fonction de x_s ,
- remplacer x_e dans les expressions du coût z et des variables de base x_i , $i \in B' = B - \{s\} + \{e\}$

On obtient les formules de pivotage.

3.1.2 Formules de pivotage

Expression de x_e en fonction de x_s

- x_s est dans l'ancienne base B : $i=s \in B \rightarrow x_s = \bar{b}_s - \sum_{j \in N} \bar{a}_{sj} x_j$

$$x_s = \bar{b}_s - \sum_{j \in N} \bar{a}_{sj} x_j = \bar{b}_s - \bar{a}_{se} x_e - \sum_{j \in N - \{e\}} \bar{a}_{sj} x_j \Rightarrow x_e = \frac{\bar{b}_s}{\bar{a}_{se}} - \frac{1}{\bar{a}_{se}} x_s - \sum_{j \in N - \{e\}} \frac{\bar{a}_{sj}}{\bar{a}_{se}} x_j$$

- x_e est dans la nouvelle base B' : $i=e \in B' \rightarrow x_e = \bar{b}_e' - \sum_{j \in N'} \bar{a}_{ej}' x_j$

En identifiant les coefficients :

$$\begin{cases} \bar{b}_e' = \frac{\bar{b}_s}{\bar{a}_{se}} \\ \bar{a}_{es}' = \frac{1}{\bar{a}_{se}} \rightarrow j = s \\ \bar{a}_{ej}' = \frac{\bar{a}_{sj}}{\bar{a}_{se}} \rightarrow j \in N - \{e\} \end{cases}$$

On exprime ensuite les autres variables de base $x_i, i \in B - \{s\}$ en remplaçant x_e .

3.1.2 Formules de pivotage

Expression des autres variables de base

- x_i est dans l'ancienne base B : $i \in B - \{s\} \rightarrow x_i = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} x_j$

$$x_i = \bar{b}_i - \bar{a}_{ie} x_e - \sum_{j \in N - \{e\}} \bar{a}_{ij} x_j = \bar{b}_i - \bar{a}_{ie} \left(\frac{1}{\bar{a}_{se}} \bar{b}_s - \frac{1}{\bar{a}_{se}} x_s - \sum_{j \in N - \{e\}} \frac{\bar{a}_{sj}}{\bar{a}_{se}} x_j \right) - \sum_{j \in N - \{e\}} \bar{a}_{ij} x_j$$

$$\Rightarrow x_i = \bar{b}_i - \frac{\bar{a}_{ie}}{\bar{a}_{se}} \bar{b}_s + \frac{\bar{a}_{ie}}{\bar{a}_{se}} \bar{b}_s x_s - \sum_{j \in N - \{e\}} \left(\bar{a}_{ij} - \frac{\bar{a}_{ie}}{\bar{a}_{se}} \bar{a}_{sj} \right) x_j$$

- x_i reste dans la nouvelle base B' : $i \in B' \rightarrow x_i = \bar{b}_i' - \sum_{j \in N'} \bar{a}_{ij}' x_j$

En identifiant les coefficients :

$$\begin{cases} \bar{b}_i' = \bar{b}_i - \frac{\bar{a}_{ie}}{\bar{a}_{se}} \bar{b}_s \\ \bar{a}_{is}' = -\frac{\bar{a}_{ie}}{\bar{a}_{se}} \rightarrow j = s \\ \bar{a}_{ij}' = \bar{a}_{ij} - \frac{\bar{a}_{ie}}{\bar{a}_{se}} \bar{a}_{sj} \rightarrow j \in N - \{e\} \end{cases}$$

3.1.2 Formules de pivotage

Expression du coût

- Dans l'ancienne base B : $z = \bar{z} + \sum_{j \in N} \bar{c}_j x_j$

$$z = \bar{z} + \bar{c}_e x_e + \sum_{j \in N - \{e\}} \bar{c}_j x_j = \bar{z} + \bar{c}_e \left(\frac{1}{\bar{a}_{se}} \bar{b}_s - \frac{1}{\bar{a}_{se}} x_s - \sum_{j \in N - \{e\}} \frac{\bar{a}_{sj}}{\bar{a}_{se}} x_j \right) + \sum_{j \in N - \{e\}} \bar{c}_j x_j$$

$$\Rightarrow z = \bar{z} + \bar{c}_e \frac{\bar{b}_s}{\bar{a}_{se}} - \frac{\bar{c}_e}{\bar{a}_{se}} x_s + \sum_{j \in N - \{e\}} \left(\bar{c}_j - \bar{c}_e \frac{\bar{a}_{sj}}{\bar{a}_{se}} \right) x_j$$

- Dans la nouvelle base B' : $z = \bar{z}' + \sum_{j \in N'} \bar{c}_j' x_j$

En identifiant les coefficients :

$$\begin{cases} \bar{z}' = \bar{z} + \bar{c}_e \frac{\bar{b}_s}{\bar{a}_{se}} \\ \bar{c}_s' = -\frac{\bar{c}_e}{\bar{a}_{se}} \rightarrow j = s \\ \bar{c}_j' = \bar{c}_j - \bar{c}_e \frac{\bar{a}_{sj}}{\bar{a}_{se}} \rightarrow j \in N - \{e\} \end{cases}$$

3.1.2 Formules de pivotage

Récapitulatif

- Nouvelles variables de base : $i \in B' = B - \{s\} + \{e\}$

$$i = e \rightarrow \begin{cases} \bar{b}_e' = \frac{\bar{b}_s}{\bar{a}_{se}} \\ \bar{a}_{es}' = \frac{1}{\bar{a}_{se}} \rightarrow j = s \\ \bar{a}_{ej}' = \frac{\bar{a}_{sj}}{\bar{a}_{se}} \rightarrow j \in N - \{e\} \end{cases} \quad i \in B - \{s\} \rightarrow \begin{cases} \bar{b}_i' = \bar{b}_i - \frac{\bar{a}_{ie}}{\bar{a}_{se}} \bar{b}_s \\ \bar{a}_{is}' = -\frac{\bar{a}_{ie}}{\bar{a}_{se}} \rightarrow j = s \\ \bar{a}_{ij}' = \bar{a}_{ij} - \frac{\bar{a}_{ie}}{\bar{a}_{se}} \bar{a}_{sj} \rightarrow j \in N - \{e\} \end{cases}$$

- Nouveau coût

$$\begin{cases} \bar{z}' = \bar{z} + \bar{c}_e \frac{\bar{b}_s}{\bar{a}_{se}} \\ \bar{c}_s' = -\frac{\bar{c}_e}{\bar{a}_{se}} \rightarrow j = s \\ \bar{c}_j' = \bar{c}_j - \bar{c}_e \frac{\bar{a}_{sj}}{\bar{a}_{se}} \rightarrow j \in N - \{e\} \end{cases}$$

On dispose dans un tableau les éléments nécessaires au pivotage → **tableau du simplexe**.

3.1.2 Méthode des tableaux

Tableau du simplexe

- On écrit le problème sous forme canonique dans la base B :

$$\min_{x_N \geq 0} z = \bar{z} + \bar{c}_N^T x_N \quad \text{sous} \quad x_B = \bar{b} - B^{-1} N x_N \geq 0 \quad \text{avec} \quad \begin{cases} \bar{b} = B^{-1} b \\ \bar{z} = c_B^T \bar{b} \\ \bar{c}_N^T = c_N^T - c_B^T B^{-1} N \end{cases}$$

- La solution de base associée à B est : $\begin{cases} x_B = \bar{b} \\ x_N = 0 \\ z = \bar{z} \end{cases}$

- Le **tableau du simplexe** est : $T = \begin{array}{c|c} B^{-1}A & \bar{b} \\ \hline \bar{c}^T & -\bar{z} \end{array} = \begin{array}{cccccc|c} & x_1 & & x_j & & x_n & \\ \hline B^{-1}A_1 & \cdots & B^{-1}A_j & \cdots & B^{-1}A_n & & x_B \\ \hline \bar{c}_1 & \cdots & \bar{c}_j & \cdots & \bar{c}_n & & -c_B^T x_B \end{array}$

$A_j = j^{\text{ème}}$ colonne de A

$AE = (B \quad N)$ avec $E =$ matrice de permutation de colonnes $\Rightarrow B^{-1}A = (I \quad B^{-1}N) E^T$

- En permutant les colonnes : $T = \begin{array}{cc|c} & x_B & x_N & \\ \hline I & B^{-1}N & & \bar{b} \\ \hline 0 & \bar{c}_N^T & & -\bar{z} \end{array} \rightarrow \begin{cases} x_B + B^{-1}N x_N = \bar{b} \\ \bar{c}_N^T x_N = z - \bar{z} \end{cases}$

3.1.2 Méthode des tableaux

Description du tableau

Le tableau du simplexe est noté $T(i,j)$: $i=1$ à $m+1$, $j=1$ à $n+1$

- $T(1:m,1:n)$: **Matrice $B^{-1}A$** $\rightarrow m \times n$

$$B^{-1}A = \begin{pmatrix} I & B^{-1}N \end{pmatrix} E^T \text{ en plaçant les variables de base en premier}$$

- $T(m+1,1:n)$: **Coûts réduits** $\rightarrow 1 \times n$

$$\bar{c}^T = c - c_B^T B^{-1}A \Rightarrow \begin{cases} \bar{c}_B^T = 0 \\ \bar{c}_N^T = c_N^T - c_B^T B^{-1}N \end{cases}$$

- $T(1:m,n+1)$: **Solution de base** $\rightarrow m \times 1$

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} \bar{b} \\ 0 \end{pmatrix}$$

- $T(m+1,n+1)$: **Opposé du coût** $\rightarrow 1 \times 1$

$$-z = -\bar{z} = -\bar{c}_B^T \bar{b}$$

x_1	\dots	x_j	\dots	x_n	
\vdots		\vdots		\vdots	\bar{b}_1
$B^{-1}A_1$	\dots	$B^{-1}A_j$	\dots	$B^{-1}A_n$	\vdots
\vdots		\vdots		\vdots	\bar{b}_m
\bar{c}_1	\dots	\bar{c}_j	\dots	\bar{c}_n	$-\bar{z}$



$$T = \begin{array}{cc|c} & x_B & x_N & \\ \hline & I & B^{-1}N & \bar{b} \\ \hline 0 & & \bar{c}_N^T & -\bar{z} \end{array} \rightarrow \begin{array}{l} x_B + B^{-1}N x_N = \bar{b} \\ \bar{c}_N^T x_N = z - \bar{z} \end{array}$$

3.1.2 Méthode des tableaux

Utilisation du tableau

Le tableau du simplexe permet de :

- Repérer les **variables de base** → colonnes = matrice identité, coûts réduits nuls
- Vérifier si la base est **admissible** → valeurs positives ou nulles des variables de base
- Vérifier si la base est **optimale** → valeurs strictement positives des coûts réduits
- Sélectionner un **pivotage** pour passer à une base adjacente meilleure
- Mettre à jour la forme canonique dans la **nouvelle base**

Méthode de pivotage

- On choisit une variable hors base de coût réduit négatif → **colonne e**
- On examine la variation des variables de base suivant la direction d_e

$$x_i = \bar{b}_i - \bar{a}_{ie} x_e, i \in B \quad \text{s'annule pour :} \quad x_e = \frac{\bar{b}_i}{\bar{a}_{ie}}$$

- La première variable de base à s'annuler sort de la base → **ligne s**
- Le **pivotage e-s** consiste à faire apparaître une colonne de la matrice identité en colonne e
 - forme canonique dans la nouvelle base
 - par combinaison linéaire des lignes du tableau

3.1.2 Méthode des tableaux


Réalisation du pivotage

La variable x_e entre dans la nouvelle base $B' = B - \{s\} + \{e\}$

Pour faire apparaître une colonne de la matrice identité en colonne e , on réalise des **combinaisons linéaires des lignes du tableau**, y compris la dernière colonne.

- Division de la ligne s par le **pivot** $\bar{a}_{se} \rightarrow \bar{a}_{se}' = 1$
- Addition de la ligne s aux autres lignes pour annuler les coefficients dans la colonne e
- Annulation du coût réduit dans la colonne e

x_1	x_e	x_n	
\vdots	\bar{a}_{1e}	\vdots	\bar{b}_1
\vdots		\vdots		\vdots	\vdots
\vdots	\bar{a}_{se}	\vdots	\bar{b}_s
\vdots		\vdots		\vdots	\vdots
\vdots	\bar{a}_{me}	\vdots	\bar{b}_m
\bar{c}_1	\bar{c}_e	\bar{c}_n	$-\bar{z}$



x_1	x_e	x_n	
\vdots	0	\vdots	\bar{b}_1'
\vdots		\vdots		\vdots	\vdots
\vdots	1	\vdots	\bar{b}_s'
\vdots		\vdots		\vdots	\vdots
\vdots	0	\vdots	\bar{b}_m'
\bar{c}_1'	0	\bar{c}_n'	$-\bar{z}'$

3.1.2 Méthode des tableaux

Algorithme de pivotage

1. Choix du pivot

- Variable hors base entrante $x_e = 1^{\text{er}}$ coût réduit négatif
- Pas maximal admissible pour chaque variable de base : $\alpha_i = \frac{T(i, n+1)}{T(i, e)}, i \in B, \text{ si } T(i, e) > 0$
- Variable de base sortante x_s : $\alpha_s = \min_{\substack{i \in B \\ T(i, e) > 0}} \alpha_i$

2. Réalisation du pivotage

- Pivot = $T(s, e)$
- Lignes $i=1, \dots, m+1, i \neq s$ $T(i, k) = T(i, k) - \frac{T(i, e)}{T(s, e)} T(s, k), k = 1, \dots, n+1$
- Ligne s du pivot $T(s, k) = \frac{T(s, k)}{T(s, e)}, k = 1, \dots, n+1$

→ méthode similaire à la méthode du pivot de Gauss

3.1.2 Exemple

Méthode des tableaux

- Problème linéaire à 3 variables x_1, x_2, x_3

$$\min_{x_1, x_2, x_3} -10x_1 - 12x_2 - 12x_3 \quad \text{sous} \quad \begin{cases} x_1 + 2x_2 + 2x_3 \leq 20 \\ 2x_1 + x_2 + 2x_3 \leq 20 \\ 2x_1 + 2x_2 + x_3 \leq 20 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

- **Forme standard**

→ **Variables d'écart** x_4, x_5, x_6 positives

$$\min_{x_1, x_2, x_3, x_4, x_5, x_6} -10x_1 - 12x_2 - 12x_3 \quad \text{sous} \quad \begin{cases} x_1 + 2x_2 + 2x_3 + x_4 = 20 \\ 2x_1 + x_2 + 2x_3 + x_5 = 20 \\ 2x_1 + 2x_2 + x_3 + x_6 = 20 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{cases}$$

3.1.2 Exemple

Méthode des tableaux

- Tableau du simplexe

x_1	x_2	x_3	x_4	x_5	x_6	
1	2	2	1	0	0	20
2	1	2	0	1	0	20
2	2	1	0	0	1	20
-10	-12	-12	0	0	0	0

x_4

x_5

x_6

-z

Base initiale admissible
 (x_4, x_5, x_6)

- Solution de base **non optimale** : coûts réduits négatifs (= directions de descente)
- Variable entrante: 1^{er} coût réduit négatif $\rightarrow x_1$
- Variable sortante : 1^{ère} variable de base à s'annuler $\rightarrow x_5$
- Pivot : $\bar{a}_{51} = 2$

3.1.2 Exemple

Méthode des tableaux

- **1^{er} pivotage** : entrée x_1 , sortie x_5

x_1	x_2	x_3	x_4	x_5	x_6		Pas
1	2	2	1	0	0	20	$x_4 \rightarrow s_{14}=20$
2	1	2	0	1	0	20	$x_5 \rightarrow s_{15}=10$
2	2	1	0	0	1	20	$x_6 \rightarrow s_{16}=10$
-10	-12	-12	0	0	0	0	z

x_1	x_2	x_3	x_4	x_5	x_6		
0	1.5	1	1	-0.5	0	10	x_4 Nouvelle base
1	0.5	1	0	0.5	0	10	x_1 (x_1, x_4, x_6)
0	1	-1	0	-1	1	0	x_6
0	-7	-2	0	5	0	100	z

3.1.2 Exemple

Méthode des tableaux

- 2^{ème} pivotage : entrée x_2 , sortie x_6

x_1	x_2	x_3	x_4	x_5	x_6		Pas
0	1.5	1	1	-0.5	0	10	$x_4 \rightarrow s_{24}=20/3$
1	0.5	1	0	0.5	0	10	$x_1 \rightarrow s_{21}=20$
0	1	-1	0	-1	1	0	$x_6 \rightarrow s_{26}=0$
0	-7	-2	0	5	0	100	z

x_1	x_2	x_3	x_4	x_5	x_6		
0	0	2.5	1	1	-1.5	10	x_4
1	0	1.5	0	1	-0.5	10	x_1
0	1	-1	0	-1	1	0	x_2
0	0	-9	0	-2	7	100	z

Nouvelle base
 (x_1, x_2, x_4)

3.1.2 Exemple

Méthode des tableaux

- 3^{ème} pivotage : entrée x_3 , sortie x_4

x_1	x_2	x_3	x_4	x_5	x_6		Pas
0	0	2.5	1	1	-1.5	10	$x_4 \rightarrow s_{34}=4$
1	0	1.5	0	1	-0.5	10	$x_1 \rightarrow s_{31}=20/3$
0	1	-1	0	-1	1	0	$x_2 \rightarrow s_{32}=+\infty$
0	0	-9	0	-2	7	100	z

x_1	x_2	x_3	x_4	x_5	x_6		
0	0	1	0.4	0.4	-0.6	4	x_3
1	0	0	-0.6	0.4	0.4	4	x_1
0	1	0	0.4	-0.6	0.4	4	x_2
0	0	0	3.6	1.6	1.6	136	z

Nouvelle base
(x_1, x_2, x_3)

- Solution optimale : $\bar{c} \geq 0 \rightarrow \begin{cases} x^* = (4 \ 4 \ 4 \ 0 \ 0 \ 0)^T \\ z^* = -136 \end{cases}$

3.1.2 Exemple

Méthode des tableaux

Récapitulatif des itérations

k	B	c			x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	d _B			s _{max}	e	s	z
0	4 5 6	-10	-12	-12	0	0	0	20	20	20	-1	-2	-2	10	1	5	0
1	1 4 6	-7	-2	5	10	0	0	10	0	0	-1.5	-0.5	-1	0	2	6	-100
2	1 2 4	-9	-2	7	10	0	0	10	0	0	-2.5	-1.5	1	4	3	4	-100
3	1 2 3	3.6	31.6	1.6	4	4	4	0	0	0							-136

Commentaires

- La mise sous forme standard nécessite d'introduire des variables supplémentaires
 → Variables d'écart positives
- On dispose directement d'une base initiale admissible formée des variables d'écart
 → Ce n'est pas toujours le cas
 → Phase préliminaire pour construire une base initiale
- Certains pivotages ne réduisent pas le coût (exemple : pivotage numéro 2)
 → Base dégénérée + risque de cyclage (= retrouver une base précédente)
- La solution optimale ne comporte que les variables initiales
 → Ce n'est pas toujours le cas
 → Des pivotages supplémentaires peuvent être nécessaires

3.1.3 Initialisation

- ☐ Problème auxiliaire
- ☐ Tableau initial
- ☐ Méthode des 2 phases

3.1.3 Problème auxiliaire

Base initiale

Le problème (PL) sous forme standard est : $\min_{x \in \mathbb{R}^n} c^T x$ sous $\begin{cases} Ax = b \\ x \geq 0 \end{cases}$

L'algorithme du simplexe nécessite une **base initiale admissible**.

Pour trouver une solution admissible, on considère un problème auxiliaire.

Problème auxiliaire

Le problème auxiliaire (PLa) sous forme standard est :

$$\min_{\substack{x \in \mathbb{R}^n \\ y \in \mathbb{R}^m}} 0^T x + e^T y \text{ sous } \begin{cases} Ax + y = b \\ x, y \geq 0 \end{cases}$$

- Les variables du problème auxiliaire sont :
 - les n variables x du problème initial (PL)
 - m variables auxiliaires $y \rightarrow 1$ variable par contrainte
- La fonction coût du problème auxiliaire est positive, bornée inférieurement par 0 (car $y \geq 0$).
- Si x_0 est un **point admissible de (PL)**, alors $(x=x_0, y=0)$ est une solution de (PLa) à coût nul
donc une solution optimale de (PLa)
On peut donc trouver un point admissible en résolvant le problème auxiliaire (PLa).

3.1.3 Problème auxiliaire

Problème auxiliaire

Le problème auxiliaire (PLa) est sous forme standard : $\min_{\substack{x \in \mathbb{R}^n \\ y \in \mathbb{R}^m}} 0^T x + e^T y$ sous $\begin{cases} Ax + y = b \\ x, y \geq 0 \end{cases}$

- On peut supposer $b \geq 0$ (au besoin en multipliant les contraintes égalité par -1).

Tableau initial du problème auxiliaire

- Le problème (PLa) est **sous forme canonique dans la base B associée aux variables y** :
 - variables de base y \rightarrow matrice $B=I$, coûts réduits=1
 - variables hors base x \rightarrow matrice $N=A$, coûts réduits=0
- La solution de base associée à la base B ($x=0, y=b$) est admissible : $y=b \geq 0$
- On peut appliquer l'algorithme du simplexe au problème (PLa) à partir de la base B.

- Tableau initial du simplexe pour le problème (PLa) :

x			y
A	I	b	y
0^T	e^T	0	-z

3.1.3 Problème auxiliaire

Solution du problème auxiliaire

S'il existe un point admissible pour (PL), alors :

- le coût optimal du problème auxiliaire (PLa) est nul,
- la solution (x_0, y_0) de (PLa) donne un point x_0 admissible du problème (PL)

$$\begin{cases} 0^T x_0 + e^T y_0 = 0 \\ Ax_0 + y_0 = b \\ x_0, y_0 \geq 0 \end{cases} \Rightarrow \begin{cases} y_0 = 0 \\ Ax_0 = b \\ x_0 \geq 0 \end{cases} \rightarrow \text{contraintes du problème (PL)}$$

Par contraposée, si le coût optimal de (PLa) n'est pas nul, (PL) n'a pas de point admissible.

Variables auxiliaires

- Les variables y sont nulles à l'optimum. Elles sont :
 - soit hors base
 - soit en base (solution dégénérée)
- Pour obtenir une base admissible du problème (PL) qui ne contienne que des variables x , il faut échanger les variables y en base par des pivotages avec des variables x . Ces pivotages sont de pas nul, le tableau n'est pas modifié.
- On obtient une **base B ne contenant que des variables x** du problème (PL).

3.1.3 Exemple

Problème auxiliaire

- Problème linéaire à 4 variables x_1, x_2, x_3, x_4

$$\min_{x_1, x_2, x_3, x_4} x_1 + x_2 + x_3 \text{ sous } \begin{cases} x_1 + 2x_2 + 3x_3 = 3 \\ -x_1 + 2x_2 + 6x_3 = 2 \\ 4x_2 + 9x_3 = 5 \\ 3x_3 + x_4 = 1 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

- **Problème auxiliaire**

→ **Variables auxiliaires** y_1, y_2, y_3, y_4 positives

$$\min_{\substack{x_1, x_2, x_3, x_4 \\ y_1, y_2, y_3, y_4}} y_1 + y_2 + y_3 + y_4 \text{ sous } \begin{cases} x_1 + 2x_2 + 3x_3 + y_1 = 3 \\ -x_1 + 2x_2 + 6x_3 + y_2 = 2 \\ 4x_2 + 9x_3 + y_3 = 5 \\ 3x_3 + x_4 + y_4 = 1 \\ x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4 \geq 0 \end{cases}$$

3.1.3 Exemple

Problème auxiliaire

- Tableau initial du simplexe pour le problème auxiliaire

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
1	2	3	0	1	0	0	0	3	y_1
-1	2	6	0	0	1	0	0	2	y_2
0	4	9	0	0	0	1	0	5	y_3
0	0	3	1	0	0	0	1	1	y_4
0	-8	-21	-1	0	0	0	0	-11	-z

- Solution de base **non optimale** : coûts réduits négatifs (= directions de descente)
- Variable entrante
 1^{er} coût réduit négatif $\rightarrow x_2$
- Variable sortante
 1^{ère} variable de base à s'annuler $\rightarrow y_2$

3.1.3 Exemple

Problème auxiliaire

- Itérations du simplexe pour le problème auxiliaire

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
1	2	3	0	1	0	0	0	3	y_1
-1	2	6	0	0	1	0	0	2	y_2
0	4	9	0	0	0	1	0	5	y_3
0	0	3	1	0	0	0	1	1	y_4
0	-8	-21	-1	0	0	0	0	-11	-z

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
2	0	-3	0	1	-1	0	0	1	y_1
-1/2	1	3	0	0	1/2	0	0	1	x_2
2	0	-3	0	0	-2	1	0	1	y_3
0	0	3	1	0	0	0	1	1	y_4
-4	0	3	-1	0	4	0	0	-3	-z

3.1.3 Exemple

Problème auxiliaire

- Itérations du simplexe pour le problème auxiliaire

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
2	0	-3	0	1	-1	0	0	1	y_1
-1/2	1	3	0	0	1/2	0	0	1	x_2
2	0	-3	0	0	-2	1	0	1	y_3
0	0	3	1	0	0	0	1	1	y_4
-4	0	3	-1	0	4	0	0	-3	-z

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
1	0	-3/2	0	1/2	-1/2	0	0	1/2	x_1
0	1	9/4	0	1/4	1/4	0	0	5/4	x_2
0	0	0	0	-1	-1	1	0	0	y_3
0	0	3	1	0	0	0	1	1	y_4
0	0	-3	-1	2	2	0	0	-1	-z

3.1.3 Exemple

Problème auxiliaire

- Itérations du simplexe pour le problème auxiliaire

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
1	0	-3/2	0	1/2	-1/2	0	0	1/2	x_1
0	1	9/4	0	1/4	1/4	0	0	5/4	x_2
0	0	0	0	-1	-1	1	0	0	y_3
0	0	3	1	0	0	0	1	1	y_4
0	0	-3	-1	2	2	0	0	-1	-z

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
1	0	0	1/2	1/2	-1/2	0	1/2	1	x_1
0	1	0	-3/4	1/4	1/4	0	-3/4	1/2	x_2
0	0	0	0	-1	-1	1	0	0	y_3
0	0	1	1/3	0	0	0	1/3	1/3	x_3
0	0	0	0	2	2	0	1	0	-z

3.1.3 Exemple

Problème auxiliaire

- Echange des variables auxiliaires y en base avec des variables x

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
1	0	0	1/2	1/2	-1/2	0	1/2	1	x_1
0	1	0	-3/4	1/4	1/4	0	-3/4	1/2	x_2
0	0	0	0	-1	-1	1	0	0	y_3
0	0	1	1/3	0	0	0	1/3	1/3	x_3
0	0	0	0	2	2	0	1	0	-z

- Tous les pivots sont nuls sur la ligne 3
 - La contrainte 3 est **redondante** (= somme des 2 premières contraintes)
 - La matrice A n'est pas de rang plein
- La procédure permet d'identifier les contraintes redondantes
 - Suppression de la contrainte 3
 - Suppression de la variable auxiliaire y_3

3.1.3 Exemple

Problème auxiliaire

- Tableau solution du problème auxiliaire

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4		
1	0	0	1/2	1/2	-1/2	0	1/2	1	x_1
0	1	0	-3/4	1/4	1/4	0	-3/4	1/2	x_2
0	0	0	0	-1	-1	1	0	0	y_3
0	0	1	1/3	0	0	0	1/3	1/3	x_3
0	0	0	0	2	2	0	1	0	-z

- Base x_1, x_2, x_3

x_1	x_2	x_3	x_4	y_1	y_2	y_4	
1	0	0	1/2	1/2	-1/2	1/2	x_1
0	1	0	-3/4	1/4	1/4	-3/4	x_2
0	0	1	1/3	0	0	1/3	x_3
0	0	0	0	2	2	1	-z

→ coût nul

→ point admissible du problème initial

3.1.3 Tableau initial

Tableau solution du problème (PLa)

- La base B solution de (PLa) est composée uniquement de variables x du problème initial (PL).
- Le tableau correspondant à la solution de (PLa) est :
 $\times \rightarrow$ valeurs à modifier pour passer au problème (PL)

x_B	x_N	y		
I	$B^{-1}N$	\times	$B^{-1}b$	x_B
0	\times	\times	\times	$-z$

Tableau initial du problème (PL)

Pour construire le tableau du simplexe du problème (PL)

- On supprime les colonnes correspondant aux variables y :
- On calcule les **coûts réduits** dans la base B : $\bar{c} = c - c_B^T B^{-1} A = \begin{cases} c_j - c_B^T B^{-1} A_j & \text{si } j \in N \\ 0 & \text{si } j \in B \end{cases}$

x_B	x_N		
I	$B^{-1}N$	$B^{-1}b$	x_B
0	\times	\times	$-Z$

On calcule le **coût** dans la base B : $z = c_B^T B^{-1} b$

Pour faciliter les calculs, on rappelle les coûts sur la 1^{ère} ligne.

c_B		c_N		
I	$B^{-1}N$	$B^{-1}b$	x_B	
0	$c_N - c_B^T B^{-1}N$	$-c_B^T B^{-1}b$	-Z	

x_B	x_N		
I	$B^{-1}N$	$B^{-1}b$	x_B
0	\bar{c}_N	-Z	-Z

3.1.3 Méthode des 2 phases

Prétraitement

- Mettre le problème sous forme standard
- Prémultiplier les contraintes (2nd membre positif)

$$\min_{x \in \mathbb{R}^n} c^T x \text{ sous } \begin{cases} Ax = b, b \geq 0 \\ x \geq 0 \end{cases}$$

Phase 1 : Problème auxiliaire

- Introduire une variable auxiliaire y par contrainte

$$\min_{\substack{x \in \mathbb{R}^n \\ y \in \mathbb{R}^m}} 0^T x + e^T y \text{ sous } \begin{cases} Ax + y = b \\ x, y \geq 0 \end{cases}$$

- Construire le tableau initial du problème auxiliaire

		x	y	
	A	I	b	y
	-e ^T A	0	-e ^T b	-z

- Résoudre le problème auxiliaire
- Faire sortir les variables auxiliaires de la base
- Supprimer les contraintes redondantes (pivots tous nuls sur une ligne)
- Supprimer les colonnes correspondant aux variables auxiliaires y
 → Base ne contenant que des variables x

Phase 2 : Problème initial

- Calculer la dernière ligne du tableau (coûts réduits, coût)
- Résoudre le problème initial

		x _B	x _N	
	I	B ⁻¹ N	B ⁻¹ b	y
	0	c _N - c _B ^T B ⁻¹ N	-c _B ^T B ⁻¹ b	-z

3.1.3 Exemple

Méthode des 2 phases

- Problème linéaire à 5 variables x_1, x_2, x_3, x_4, x_5

$$\min_{x_1, x_2, x_3, x_4, x_5} 2x_1 + 3x_2 + 3x_3 + x_4 - 2x_5 \text{ sous } \begin{cases} x_1 + 3x_2 + 4x_4 + x_5 = 2 \\ x_1 + 2x_2 - 3x_4 + x_5 = 2 \\ -x_1 - 4x_2 + 3x_3 = 1 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases}$$

- **Problème auxiliaire**

→ **Variables auxiliaires** y_1, y_2, y_3 positives

$$\min_{\substack{x_1, x_2, x_3, x_4, x_5 \\ y_1, y_2, y_3}} y_1 + y_2 + y_3 \text{ sous } \begin{cases} x_1 + 3x_2 + 4x_4 + x_5 + y_1 = 2 \\ x_1 + 2x_2 - 3x_4 + x_5 + y_2 = 2 \\ -x_1 - 4x_2 + 3x_3 + y_3 = 1 \\ x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3 \geq 0 \end{cases}$$

3.1.3 Exemple

Méthode des 2 phases

- Tableau initial du problème auxiliaire

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3		
1	3	0	4	1	1	0	0	2	y_1
1	2	0	-3	1	0	1	0	2	y_2
-1	-4	3	0	0	0	0	1	1	y_3
-1	-1	-3	-1	-2	0	0	0	-5	-z

- Solution de base **non optimale** : coûts réduits négatifs (= directions de descente)
- Variable entrante
 1^{er} coût réduit négatif $\rightarrow x_1$
- Variable sortante
 1^{ère} variable de base à s'annuler $\rightarrow y_1$

3.1.3 Exemple

Méthode des 2 phases

- Itérations du problème auxiliaire

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3		
1	3	0	4	1	1	0	0	2	y_1
1	2	0	-3	1	0	1	0	2	y_2
-1	-4	3	0	0	0	0	1	1	y_3
-1	-1	-3	-1	-2	0	0	0	-5	-z

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3		
1	3	0	4	1	1	0	0	2	x_1
0	-1	0	-7	0	-1	1	0	0	y_2
0	-1	3	4	1	1	0	1	3	y_3
0	2	-3	3	-1	1	0	0	-3	-z

3.1.3 Exemple

Méthode des 2 phases

- Itérations du problème auxiliaire

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3		
1	3	0	4	1	1	0	0	2	x_1
0	-1	0	-7	0	-1	1	0	0	y_2
0	-1	3	4	1	1	0	1	3	y_3
0	2	-3	3	-1	1	0	0	-3	-z

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3		
1	3	0	4	1	1	0	0	2	x_1
0	-1	0	-7	0	-1	1	0	0	y_2
0	-1/3	1	4/3	1/3	1/3	0	1/3	1	x_3
0	1	0	7	0	2	0	1	0	-z

- Solution optimale du problème auxiliaire (coûts réduits positifs ou nuls)

3.1.3 Exemple

Méthode des 2 phases

- Base initiale : échange $y_2 - x_2$

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3		
1	3	0	4	1	1	0	0	2	x_1
0	-1	0	-7	0	-1	1	0	0	y_2
0	-1/3	1	4/3	1/3	1/3	0	1/3	1	x_3
0	1	0	7	0	2	0	1	0	-z

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3		
1	0	0	-17	1	2	3	0	2	x_1
0	1	0	7	0	-1	-1	0	0	x_2
0	0	1	3.67	1/3	2/3	-1/3	1/3	1	x_3
0	0	0	0	0	1	1	1	0	-z

3.1.3 Exemple

Méthode des 2 phases

- Base initiale : **suppression variables auxiliaires**

x_1	x_2	x_3	x_4	x_5	
1	0	0	-17	1	2
0	1	0	7	0	0
0	0	1	3.67	1/3	1
0	0	0	0	0	0

x_1

x_2

x_3

$-Z$

- Calcul de la dernière ligne du tableau

On rappelle les coûts en 1^{ère} ligne pour faciliter le calcul (multiplication par les colonnes)

x_1	x_2	x_3	x_4	x_5	
2	3	3	1	-2	
1	0	0	-17	1	2
0	1	0	7	0	0
0	0	1	3.67	1/3	1
0	0	0	3	-5	-7

x_1

x_2

x_3

$-Z$

$$\begin{aligned} \text{coûts réduits} &= c_N - c_B^T B^{-1} N \\ \text{coût} &= -c_B^T B^{-1} b \end{aligned}$$

3.1.3 Exemple

Méthode des 2 phases

- Itérations du problème initial

x_1	x_2	x_3	x_4	x_5	
1	0	0	-17	1	2
0	1	0	7	0	0
0	0	1	3.67	1/3	1
0	0	0	3	-5	-7

x_1
 x_2
 x_3
 $-z$

x_1	x_2	x_3	x_4	x_5	
1	0	0	-17	1	2
0	1	0	7	0	0
-1/3	0	1	9.33	0	1/3
5	0	0	-82	0	3

x_5
 x_2
 x_3
 $-z$

3.1.3 Exemple

Méthode des 2 phases

- Itérations du problème initial

x_1	x_2	x_3	x_4	x_5	
1	0	0	-17	1	2
0	1	0	7	0	0
-1/3	0	1	9.33	0	1/3
5	0	0	-82	0	3

x_5

x_2

x_3

$-z$

x_1	x_2	x_3	x_4	x_5	
1	2.43	0	0	1	2
0	0.14	0	1	0	0
-1/3	-1.33	1	0	0	1/3
5	11.71	0	0	0	3

x_5

x_4

x_3

$-z$

- **Solution optimale** : $\bar{c} \geq 0 \rightarrow \begin{cases} x^* = (0 \ 0 \ 1/3 \ 0 \ 2)^T \\ z^* = -3 \end{cases}$

3.1.4 Simplexe révisé

Méthode révisée du simplexe

- L'algorithme du simplexe nécessite un grand nombre d'opérations matricielles
 - problèmes de temps de calcul
 - problèmes de place mémoire
 - problèmes de précision numérique
- La méthode révisée du simplexe permet de réduire le nombre d'opérations.

Rappels

- Forme standard :
$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{sous} \quad \begin{cases} Ax = b & \rightarrow \lambda \\ x \geq 0 & \rightarrow s \end{cases}$$
- Multiplicateurs :
$$\begin{cases} \lambda = B^{-T} c_B \\ s_N = c_N - (B^{-1} N)^T c_B = \bar{c}_N \end{cases}$$
- Forme canonique dans la base B :
$$\min_{x_N \geq 0} z = \bar{z} + \bar{c}_N^T x_N \quad \text{sous} \quad x_B = \bar{b} - B^{-1} N x_N \geq 0$$
- Solution de base associée à B :
$$\begin{cases} x_B = \bar{b} \\ x_N = 0 \\ z = \bar{z} \end{cases} \quad \text{avec} \quad \begin{cases} \bar{b} = B^{-1} b \\ \bar{z} = c_B^T \bar{b} \\ \bar{c}_N^T = c_N^T - c_B^T B^{-1} N \end{cases}$$

3.1.4 Simplexe révisé

Itération du simplexe

- On connaît les colonnes de A correspondant à la base courante admissible → matrice B
 Pour réaliser une itération du simplexe, on doit déterminer :
 - la solution de base courante
 - les coûts réduits des variables hors base pour choisir la variable entrante
 - les pas maximaux sur les variables de base pour choisir la variable sortante

- Solution de base :**
$$\begin{cases} x_B = \bar{b} = B^{-1}b \\ x_N = 0 \end{cases} \quad \rightarrow Bx_B = b$$
- Coûts réduits :**
$$\bar{c}_N^T = c_N^T - c_B^T B^{-1}N = c_N^T - \lambda^T N \quad \rightarrow B\lambda = c_B$$

On choisit la variable hors base entrante x_e (coût réduit négatif)
 → colonne A_e de la matrice A

- Direction de base d_e :
$$d_e = \begin{pmatrix} d_{eB} \\ d_{eN} \end{pmatrix} \text{ avec } \begin{cases} d_{eB} = -B^{-1}A_e \\ d_{eN} = (0 \cdots 1 \cdots 0) \end{cases} \quad \rightarrow Bd_{eB} = -A_e$$

Le **pas maximal** suivant d_e correspond au rapport x_B / d_{eB} positif, minimal
 La variable de base sortante x_s est la première à s'annuler.

3.1.4 Simplexe révisé

Méthode révisée du simplexe

- Le pivotage peut être réalisé en résolvant **3 systèmes linéaires de même matrice B**.

$$\begin{cases} Bx_B = b & \rightarrow x_B \\ B\lambda = c_B & \rightarrow \lambda \rightarrow \bar{c}_N \rightarrow x_e \\ Bd_{eB} = -A_e & \rightarrow d_{eB} \rightarrow x_s \end{cases}$$

→ réduction des calculs nécessaires

→ différentes méthodes possibles d'inversion et de stockage de la matrice B

- Il suffit de stocker en mémoire :
 - **les matrices initiales A, b, c**
 - **les numéros des colonnes de base.**

→ limitation de la place mémoire
 → réduction des erreurs numériques, car on repart systématiquement des matrices initiales.
- La méthode révisée est implémentée dans les logiciels de programmation linéaire utilisant l'algorithme du simplexe.

3.1.5 Simplexe dual

Méthode du simplexe dual

L'algorithme du simplexe dual consiste à appliquer la méthode du simplexe **au problème dual**.

- **Correspondances primal (P) – dual (D) :**

Le tableau est utilisable dans les 2 sens :
 de (P) vers (D)
 ou de (D) vers (P) car le dual de (D) est (P).

Primal (P)		Dual (D)
$\min_{x \in \mathbb{R}^n} c^T x$	1	$\max_y b^T y$
$Ax = b$	m	$y \in \mathbb{R}$
$Ax \leq b$	m	$y \geq 0$
$x \geq 0$	n	$A^T y \leq c$
$x \in \mathbb{R}$	n	$A^T y = c$

- Forme canonique de (P) dans la base B

$$\min_{x_N \geq 0} z = \bar{z} + \bar{c}_N^T x_N \quad \text{sous} \quad \begin{cases} x_B + B^{-1} N x_N = \bar{b} \\ x \geq 0 \end{cases} \quad \text{avec} \quad \begin{cases} \bar{b} = B^{-1} b \\ \bar{z} = c_B^T \bar{b} \\ \bar{c}_N^T = c_N^T - c_B^T B^{-1} N \end{cases}$$

Pour appliquer la méthode du simplexe au problème dual, on doit écrire la forme canonique du problème dual dans la base B.

3.1.5 Simplexe dual

Forme canonique du dual

- On part de la forme canonique du problème primal (P) dans la base B.

$$(P) \quad \min_{x_B, x_N} \bar{z} + \bar{c}_N^T x_N \quad \text{sous} \quad \begin{cases} x_B + B^{-1} N x_N = \bar{b} & \rightarrow m \text{ contraintes} \\ x_B, x_N \geq 0 & \rightarrow n \text{ variables} \end{cases}$$

- On peut considérer les variables de base x_B comme des variables d'écart positives. On obtient un problème (P') ne portant que sur les variables hors base x_N .

$$(P') \quad \min_{x_N} \bar{c}_N^T x_N \quad \text{sous} \quad \begin{cases} B^{-1} N x_N \leq \bar{b} & \rightarrow m \text{ contraintes} \\ x_N \geq 0 & \rightarrow n-m \text{ variables} \end{cases}$$

- On écrit (P') comme un problème de maximisation, pour obtenir un problème de minimisation en passant au dual.

$$(P') \quad \max_{x_N} -\bar{c}_N^T x_N \quad \text{sous} \quad \begin{cases} B^{-1} N x_N \leq \bar{b} & \rightarrow m \text{ contraintes} \\ x_N \geq 0 & \rightarrow n-m \text{ variables} \end{cases}$$

- On passe au dual (D') de (P')
 → en utilisant le tableau de correspondances dans le sens de (D) vers (P)

3.1.5 Simplexe dual

Forme canonique du dual

- Le dual (D') de (P') s'écrit :

$$(P') \quad \max_{x_N} -\bar{c}_N^T x_N \quad \text{sous} \quad \begin{cases} B^{-1} N x_N \leq \bar{b} \\ x_N \geq 0 \end{cases} \quad \begin{array}{l} \rightarrow m \text{ contraintes} \\ \rightarrow n-m \text{ variables} \end{array}$$

$$(D') \quad \min_{y_B} \bar{b}^T y_B \quad \text{sous} \quad \begin{cases} (B^{-1} N)^T y_B \leq -\bar{c}_N \\ y_B \geq 0 \end{cases} \quad \begin{array}{l} \rightarrow n-m \text{ contraintes} \\ \rightarrow m \text{ variables} \end{array}$$

- On met (D') sous forme standard avec des variables d'écart y_N positives. On obtient un problème (D) à n variables.

$$(D) \quad \min_{y_B, y_N} \bar{b}^T y_B \quad \text{sous} \quad \begin{cases} y_N - (B^{-1} N)^T y_B = \bar{c}_N \\ y_B, y_N \geq 0 \end{cases} \quad \begin{array}{l} \rightarrow n-m \text{ contraintes} \\ \rightarrow n \text{ variables} \end{array}$$

- Le problème (D) est sous **forme canonique dans la base B** :
 - variables de base $\rightarrow y_N$ \rightarrow notations inversées par rapport au problème primal
 - variables hors base $\rightarrow y_B$

On peut écrire le tableau simplexe pour le problème (D) et appliquer les règles de pivotage.

3.1.5 Simplexe dual

Tableau simplexe du dual

- Forme canonique de (D) dans la base B.

$$(D) \min_{y_B, y_N} \bar{b}^T y_B \quad \text{sous} \quad \begin{cases} y_N - (B^{-1}N)^T y_B = \bar{c}_N & \rightarrow n-m \text{ contraintes} \\ y_B, y_N \geq 0 & \rightarrow n \text{ variables} \end{cases}$$

- Tableau T_D du simplexe de (D) dans la base B :** $T_D =$

y_N	y_B	
I	$-(B^{-1}N)^T$	\bar{c}_N^T
0	\bar{b}	$-\bar{z}$

Variables de base : $y_N \rightarrow$ valeurs \bar{c}_N

Variables hors base : $y_B \rightarrow$ coûts réduits \bar{b}

Matrice des contraintes : $-A^T \rightarrow -(B^{-1}N)^T$

- La solution de base associée à la base B est : $\begin{cases} y_N = \bar{c}_N \\ y_B = 0 \end{cases}$
- La base B est admissible si $y_N = \bar{c}_N \geq 0 \rightarrow$ **base dual-admissible**
- On applique les règles de pivotage du simplexe.
 - variable hors base entrante : coût réduit négatif
 - variable de base sortante : première variable à s'annuler

3.1.5 Simplexe dual

Pivotage sur le tableau dual

Les notations sont inversées par rapport au problème primal

- indices B → variables hors base
- indices N → variables de base

$$T_D = \begin{array}{c|c|c} y_N & y_B & \\ \hline I & -(B^{-1}N)^T & \bar{c}_N^T \\ \hline 0 & \bar{b} & -\bar{z} \end{array}$$

1. Choix du pivot

- Variable hors base entrante $y_e = 1^{\text{er}}$ coût réduit négatif $\boxed{\bar{b}_e < 0, e \in B}$
- Pas maximal admissible pour chaque variable de base : $\alpha_i = \frac{\bar{c}_{Ni}}{-\bar{a}_{ie}}, i \in N, \text{ si } -\bar{a}_{ie} > 0$
- Variable de base sortante y_s : $\alpha_s = \min_{\substack{i \in N \\ \bar{a}_{ie} < 0}} \alpha_i$

Ligne s de la variable sortante :

$$\boxed{s \in N \rightarrow \min_{\substack{i \in N \\ \bar{a}_{ie} < 0}} \frac{\bar{c}_{Ni}}{-\bar{a}_{ie}} \Leftrightarrow \max_{\substack{i \in N \\ \bar{a}_{ie} < 0}} \frac{\bar{c}_{Ni}}{\bar{a}_{ie}}}$$

2. Réalisation du pivotage

- Pivot = $\bar{a}_{se} < 0$
- Elimination pour faire apparaître des zéros sur la colonne e du pivot

3.1.5 Simplexe dual

Pivotage sur le tableau primal

- On observe que le pivotage dual peut être réalisé à partir du tableau primal **sans écrire explicitement le tableau dual**.

$$T_D = \begin{array}{c|c|c} & y_N & y_B \\ \hline & I & -(B^{-1}N)^T \\ \hline & 0 & \bar{b} \\ \hline & & -\bar{z} \end{array}$$

→ n-m contraintes

→ n-m variables de base y_N

$$T_P = \begin{array}{c|c|c} & x_B & x_N \\ \hline & I & B^{-1}N \\ \hline & 0 & \bar{c}_N^T \\ \hline & & -\bar{z} \end{array}$$

→ m contraintes

→ m variables de base x_B

- Choisir la 1^{ère} variable de base négative x_e :

$$\bar{b}_e < 0, e \in B$$

→ ligne e

→ variable **sortante**

- Déterminer la 1^{ère} variable hors base x_s à s'annuler :

$$s \rightarrow \max_{\substack{i \in N \\ \bar{a}_{ie} < 0}} \frac{\bar{c}_{Ni}}{\bar{a}_{ie}}$$

→ colonne s

→ variable **entrante**

- Effectuer le pivotage e-s de façon usuelle.

3.1.5 Simplexe dual

Comparaison simplexe primal et dual

- L'algorithme du simplexe primal maintient une base **primal-admissible** : $\bar{b} \geq 0$
L'optimum est atteint lorsque les coûts réduits sont positifs ou nuls : $\bar{c}_N \geq 0$
- L'algorithme du simplexe dual maintient une base **dual-admissible** : $\bar{c}_N \geq 0$
L'optimum est atteint lorsque les variables de base sont positives ou nulles : $\bar{b} \geq 0$

Intérêt du simplexe dual

L'algorithme du simplexe dual est adapté si l'on dispose d'une base dual-admissible.

Ceci se produit lorsque l'on modifie un problème linéaire déjà résolu par le simplexe primal.

- en ajoutant des contraintes au problème
- en modifiant les seuils des contraintes
- en fixant des variables à une valeur différente de la solution

Ces modifications : - ne changent pas les coûts réduits ($\rightarrow \bar{c}_N \geq 0$)

- rendent certaines variables de base négatives

\rightarrow La solution de base n'est plus primal-admissible, mais reste dual-admissible.

Application : problèmes de programmation linéaire mixte (entiers et réels)

3.1.5 Exemple

Simplexe dual

- Problème linéaire à 5 variables x_1, x_2, x_3, x_4, x_5

$$\min_{x_1, x_2, x_3, x_4, x_5} x_1 + 2x_2 + 2x_3 + 3x_4 + x_5 \text{ sous } \begin{cases} x_1 + x_2 = 1 \\ -x_2 - x_3 + x_5 = 0 \\ -x_1 + x_3 + x_4 = 0 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases}$$

- On choisit comme **base initiale** (x_2, x_3, x_4) .

La solution de base associée est : $\begin{cases} x_1 = 0 \\ x_5 = 0 \end{cases} \Rightarrow \begin{cases} x_2 = 1 \\ x_3 = -1 \\ x_4 = 1 \end{cases} \rightarrow \text{base non primal admissible}$

- La matrice des contraintes est : $A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ -1 & 0 & 1 & 1 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

- Pour construire le tableau du simplexe, il faut mettre le problème sous **forme canonique** dans la base (x_2, x_3, x_4) en faisant apparaître des zéros par élimination dans les colonnes 2, 3 et 4.

3.1.5 Exemple

Simplexe dual

- Problème linéaire à 5 variables x_1, x_2, x_3, x_4, x_5

$$\min_{x_1, x_2, x_3, x_4, x_5} x_1 + 2x_2 + 2x_3 + 3x_4 + x_5 \text{ sous } \begin{cases} x_1 + x_2 = 1 \\ -x_2 - x_3 + x_5 = 0 \\ -x_1 + x_3 + x_4 = 0 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases}$$

- Tableau de départ

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
0	-1	-1	0	1	0
-1	0	1	1	0	0
1	2	2	3	1	0

→ contraintes

→ coût

- On fait apparaître :
 - une matrice identité sur les colonnes de x_2, x_3, x_4
 - des zéros sur les coûts de x_2, x_3, x_4

3.1.5 Exemple

Simplexe dual

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
0	-1	-1	0	1	0
-1	0	1	1	0	0
1	2	2	3	1	0

Elimination x_2

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
1	0	-1	0	1	1
-1	0	1	1	0	0
-1	0	2	3	1	-2

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
1	0	-1	0	1	1
-1	0	1	1	0	0
-1	0	2	3	1	-2

Elimination x_3

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
-1	0	1	0	-1	-1
0	0	0	1	1	1
1	0	0	3	3	0

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
-1	0	1	0	-1	-1
0	0	0	1	1	1
1	0	0	3	3	0

Elimination x_4

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
-1	0	1	0	-1	-1
0	0	0	1	1	1
1	0	0	0	0	-3

3.1.5 Exemple

Simplexe dual

- Tableau du simplexe dans la base (x_2, x_3, x_4) .

	x_1	x_2	x_3	x_4	x_5	
$c =$	1	2	2	3	1	
	1	1	0	0	0	1
	-1	0	1	0	-1	-1
	0	0	0	1	1	1
	1	0	0	0	0	-3
						$-Z$

x_2
 x_3
 x_4

→

	c_B	c_N	
	I	$B^{-1}N$	$B^{-1}b$
	0	$c_N - c_B^T B^{-1}N$	$-c_B^T B^{-1}b$
			x_B
			$-Z$

- On vérifie bien que la dernière ligne correspond à
$$\begin{cases} \bar{c}_N = c_N - c_B^T B^{-1}N \\ -Z = -c_B^T B^{-1}b \end{cases}$$
- La base est :
 - non admissible pour le primal ($x_3 < 0$)
 - admissible pour le dual ($\bar{c}_N \geq 0$)

On peut appliquer l'algorithme dual du simplexe pour résoudre le problème.

3.1.5 Exemple

Simplexe dual

- Tableau du simplexe dans la base (x_2, x_3, x_4) .

x_1	x_2	x_3	x_4	x_5		
1	1	0	0	0	1	x_2
-1	0	1	0	-1	-1	x_3
0	0	0	1	1	1	x_4
1	0	0	0	0	-3	-Z

Base dual-admissible (x_2, x_3, x_4)

- Solution de base **non optimale** : variables de base négatives

- Variable **sortante** : 1^{ère} variable de base négative $\rightarrow x_3$

$$\bar{b}_e < 0, e \in B$$

- Variable **entrante** : 1^{er} coût réduit à s'annuler $\rightarrow x_5$

$$s \rightarrow \max_{\substack{i \in N \\ \bar{a}_{ie} < 0}} \frac{\bar{c}_{Ni}}{\bar{a}_{ie}}$$

- Pivot : $\bar{a}_{25} = -1$

3.1.5 Exemple

Simplexe dual

- 1^{er} pivotage : entrée x_5 , sortie x_3

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
-1	0	1	0	-1	-1
0	0	0	1	1	1
1	0	0	0	0	-3
					x_2
					x_3
					x_4
					$-Z$

x_1	x_2	x_3	x_4	x_5	
1	1	0	0	0	1
1	0	-1	0	1	1
-1	0	1	1	0	0
1	0	0	0	0	-3
					x_2
					x_5
					x_4
					$-Z$

Nouvelle base (x_2, x_4, x_5)

- primal-admissible

- dual-admissible

$$\bar{b} \geq 0$$

$$\bar{c}_N \geq 0$$

→ optimale

Solution : $x^* = (0 \ 1 \ 0 \ 0 \ 1)$
 $z^* = 3$

Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
- 3. Optimisation avec contraintes**
 - 3.1 Simplexe
 - 3.2 Point intérieur**
 - 3.2.1 Barrière
 - 3.2.2 Chemin central
 - 3.2.3 Algorithmes
 - 3.2.4 Extensions
 - 3.3 Gradient projeté
 - 3.4 Lagrangien augmenté
 - 3.5 Programmation quadratique séquentielle
 - 3.6 Convergence

3.2.1 Barrière

- ☐ Points intérieurs
- ☐ Fonction barrière
- ☐ Méthode barrière
- ☐ Problème linéaire
- ☐ Exemple

3.2.1 Points intérieurs

Problème avec contraintes

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \\ x \in X \end{cases}$$

Points intérieurs

- Ensemble des points admissibles : $X_{\text{adm}} = \{x \in \mathbb{R}^n / x \in X, c_E(x) = 0, c_I(x) \leq 0\}$
 - Ensemble des point intérieurs : $X_{\text{int}} = \{x \in \mathbb{R}^n / x \in X, c_E(x) = 0, c_I(x) < 0\}$
- contraintes égalité conservées (définition élargie d'un point intérieur par voisinage)
→ contraintes **inégalité strictes**

Hypothèses

- X_{int} n'est pas vide
- Tout point admissible peut être approché arbitrairement par un point intérieur

$$\forall x \in X_{\text{adm}}, \forall \varepsilon > 0, \exists \tilde{x} \in X_{\text{int}} / \|\tilde{x} - x\| \leq \varepsilon$$

→ hypothèses vérifiées dans le cas convexe (ensemble X_{int} et contraintes c_E, c_I)).

3.2.1 Fonction barrière

Fonction barrière

Une fonction $B : X_{\text{int}}$ dans \mathbb{R} est une fonction barrière si $\lim_{x \in X_{\text{int}}, c_1(x) \rightarrow 0} B(x) = +\infty$

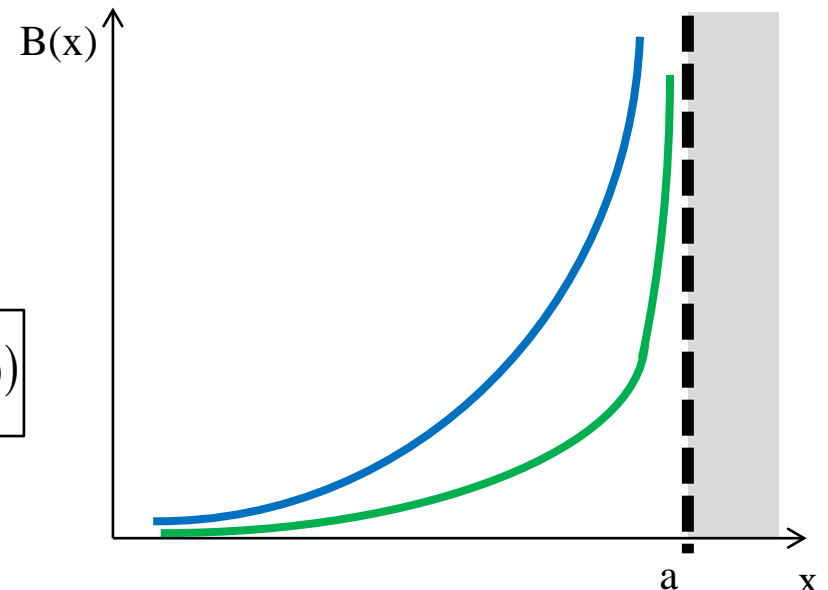
La fonction barrière tend vers l'infini lorsque l'on s'approche du bord de X_{adm}
 i.e. lorsque les contraintes inégalité $c_1(x)$ deviennent actives.

Exemple

Contrainte de borne $x < a$

Fonctions barrières usuelles

- Barrière logarithmique :
$$B(x) = - \sum_{k=1}^m \ln(c_{Ik}(x))$$
- Barrière inverse :
$$B(x) = - \sum_{k=1}^m \frac{1}{c_{Ik}(x)}$$



3.2.1 Méthode barrière

Méthode barrière

La méthode barrière consiste à combiner la fonction coût avec une fonction barrière.

La fonction barrière est pénalisée par un paramètre $h > 0 \rightarrow$ **hauteur de la barrière**

- Problème avec contraintes inégalités :
$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \\ x \in X \end{cases} \rightarrow \text{solution } x^*$$
- **Problème barrière** associé :
$$\min_{x \in \mathbb{R}^n} f_h(x) = f(x) + hB(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ x \in X \end{cases} \rightarrow \text{solution } x(h)$$

 \rightarrow Problème avec contraintes égalité plus simple (contraintes actives)
 Pour $h=0$, on retrouve le problème initial

Hauteur de la barrière

- La barrière empêche la solution $x(h)$ de s'approcher du bord du domaine admissible.
 (contraintes inégalités actives)
- On résout une suite de problèmes avec des hauteurs de barrières décroissantes
 (h_k) , $h_{k+1} < h_k$, $\lim_{k \rightarrow \infty} h_k = 0 \rightarrow$ solutions $x(h_k)$
- Méthodes peu utilisées sous cette forme \rightarrow approche des méthodes de points intérieurs
 application sur problème linéaire, puis non linéaire

3.2.1 Problème linéaire

Problème linéaire

Forme standard : $\min_{x \in \mathbb{R}^n} c^T x$ sous $\begin{cases} Ax = b \\ x \geq 0 \end{cases}$ avec $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n \rightarrow$ problème (PL)

- Ensemble des point admissibles : $X_{\text{adm}} = \{x \in \mathbb{R}^n / Ax = b, x \geq 0\} =$ polytope des contraintes
- Ensemble des point intérieurs : $X_{\text{int}} = \{x \in \mathbb{R}^n / Ax = b, x > 0\}$

Problème barrière

- Barrière logarithmique : $B(x) = -\sum_{i=1}^n \ln(x_i)$
- Problème barrière associé : $\min_{x \in \mathbb{R}^n} f_h(x) = c^T x - h \sum_{i=1}^n \ln(x_i)$ sous $\begin{cases} Ax = b \\ x > 0 \end{cases} \rightarrow$ problème (PB_h)

Solution

- $h > 0 \rightarrow$ solution x_h
- $h = 0 \rightarrow$ solution x^* du problème initial
- $h \rightarrow \infty \rightarrow$ solution $x_\infty =$ **centre analytique du polytope P**

3.2.1 Exemple

Problème linéaire

- Forme standard :
$$\min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3 \text{ sous } \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$
- Polytope des contraintes :
$$P = \left\{ (x_1, x_2, x_3) \in \mathbb{R}^3 / \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases} \right\}$$

Problème barrière

- Barrière logarithmique :
$$\min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3 - h(\ln x_1 + \ln x_2 + \ln x_3) \text{ sous } \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$
- Centre analytique x_∞** :
$$\min_{x_1, x_2, x_3} B(x_1, x_2, x_3) = -(\ln x_1 + \ln x_2 + \ln x_3) \text{ sous } \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

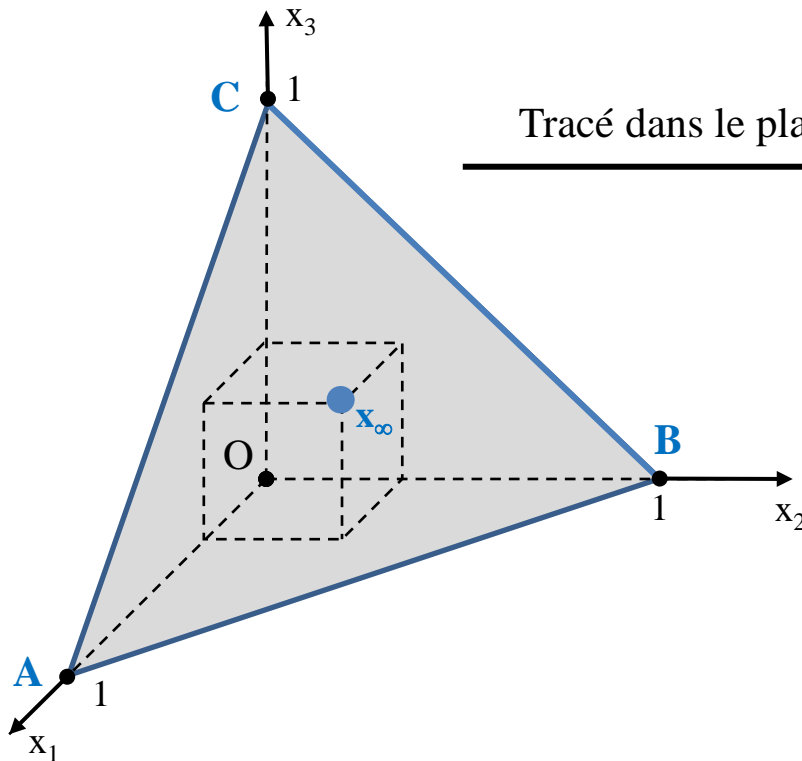
$$x_3 = 1 - x_1 - x_2 \rightarrow \min_{x_1, x_2} B(x_1, x_2) = -(\ln x_1 + \ln x_2 + \ln(1 - x_1 - x_2))$$

$$\begin{cases} \frac{\partial B}{\partial x_1} = -\frac{1}{x_1} + \frac{1}{1 - x_1 - x_2} = 0 \\ \frac{\partial B}{\partial x_2} = -\frac{1}{x_2} + \frac{1}{1 - x_1 - x_2} = 0 \end{cases} \Rightarrow \begin{cases} x_1 = \frac{1}{3} \\ x_2 = \frac{1}{3} \end{cases} \Rightarrow x_3 = \frac{1}{3} \rightarrow x_\infty = \left(\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \right)$$

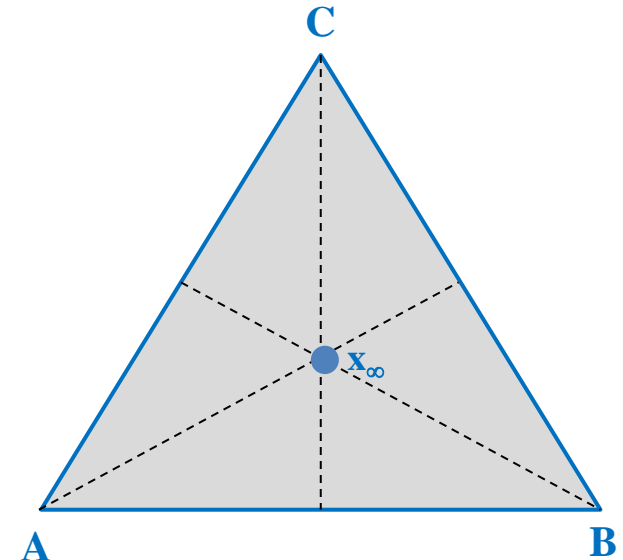
3.2.1 Exemple

Représentation graphique

- Polytope des contraintes : $P = \left\{ (x_1 \ x_2 \ x_3) \in \mathbb{R}^3 / \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases} \right\}$
- Centre analytique x_∞ : $x_\infty = \left(\frac{1}{3} \ \frac{1}{3} \ \frac{1}{3} \right)$



Tracé dans le plan (A,B,C)



3.2.2 Chemin central

- ☐ Chemin central primal
- ☐ Conditions d'optimalité
- ☐ Chemin central primal-dual
- ☐ Déplacement
- ☐ Mesure de dualité
- ☐ Voisinage
- ☐ Exemple

3.2.2 Chemin central primal

Chemin central primal

Le chemin central primal est **l'ensemble des solutions x_h** lorsque la hauteur de barrière h décroît de l'infini à 0 $\rightarrow \{x_h, h \geq 0\}$

- Début : x_∞ = centre analytique du polytope
- Fin : x^* = solution du problème linéaire

Pour construire précisément le chemin central, il faudrait résoudre l'ensemble des problèmes successifs (PB_h) sous contraintes égalité pour $h \geq 0$

$$\min_{x \in \mathbb{R}^n} f_h(x) = c^T x - h \sum_{i=1}^n \ln(x_i) \quad \text{sous} \quad \begin{cases} Ax = b \\ x > 0 \end{cases}$$

\rightarrow non réalisable en pratique (trop coûteux)

Algorithme de point intérieur

On utilise le chemin central pour définir la direction du déplacement.

On cherche à rester **au voisinage du chemin central** sans le suivre précisément.

3.2.2 Conditions d'optimalité

Problème barrière

$$\min_{x \in \mathbb{R}^n} f_h(x) = c^T x - h \sum_{i=1}^n \ln(x_i) \quad \text{sous} \quad \begin{cases} Ax = b \\ x > 0 \end{cases} \quad \begin{array}{l} \rightarrow \text{multiplicateurs } \lambda \\ \rightarrow \text{multiplicateurs } s \end{array}$$

- Lagrangien : $L(x, \lambda, s) = f_h(x) + \lambda^T (b - Ax) - s^T x$

$$= c^T x - h \sum_{i=1}^n \ln(x_i) + \lambda^T (b - Ax) - s^T x$$
- On définit les matrices diagonales X et S à partir de x et s

$$X = \begin{pmatrix} x_1 & 0 & \cdots & 0 & 0 \\ 0 & x_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & x_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & x_n \end{pmatrix} \quad S = \begin{pmatrix} s_1 & 0 & \cdots & 0 & 0 \\ 0 & s_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & s_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & s_n \end{pmatrix} \quad e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}$$

3.2.2 Conditions d'optimalité

Conditions d'optimalité

- Condition d'ordre 1 : $\nabla_x L(x, \lambda, s) = 0 \Rightarrow c - hX^{-1} - A^T \lambda - s = 0$
- Condition complémentaire : $s_i x_i = 0 \Rightarrow XS e = 0$

On définit :
$$\begin{cases} s_h = s + hX^{-1} \\ S_h = S + hX^{-1} \end{cases} \Rightarrow \begin{cases} A^T \lambda + s_h - c = 0 \\ XS_h e = h e \end{cases}$$

- Comparaison des conditions d'ordre 1 du problème barrière (PB_h) et du problème initial (PL)

Problème barrière (PB_h)

$$\begin{cases} Ax_h - b = 0 \\ A^T \lambda_h + s_h - c = 0 \\ X_h S_h = h e \\ x_h \geq 0 \\ s_h \geq 0 \end{cases} \rightarrow (x_h, \lambda_h, s_h)$$

Problème linéaire (PL)

$$\begin{cases} Ax - b = 0 \\ A^T \lambda + s - c = 0 \\ XS = 0 \\ x \geq 0 \\ s \geq 0 \end{cases} \rightarrow (x^*, \lambda^*, s^*)$$

3.2.2 Chemin central primal-dual

Chemin central primal-dual

- Les conditions d'optimalité du problème barrière (PB_h) deviennent celles du problème initial (PL) lorsque h tend vers 0.
- On cherche à résoudre le problème linéaire en considérant l'ensemble des variables **primales et duales** (x, λ, s) dans \mathbb{R}^{n+m+n}
- Ensemble admissible :
$$X_{\text{adm}} = \{(x, \lambda, s) / Ax - b = 0, A^T \lambda + s - c = 0, x \geq 0, s \geq 0\}$$
- Ensemble des points intérieurs :
$$X_{\text{int}} = \{(x, \lambda, s) / Ax - b = 0, A^T \lambda + s - c = 0, x > 0, s > 0\}$$
- Le chemin central primal-dual est l'ensemble des solutions (x_h, λ_h, s_h) lorsque la hauteur de barrière h décroît de l'infini à 0.
- Fin : $(x_0, \lambda_0, s_0) = (x^*, \lambda^*, s^*) = \text{solution du problème linéaire}$

Algorithme de point intérieur

On utilise le chemin central primal-dual pour définir la direction du déplacement.
 Le déplacement est limité pour rester dans l'ensemble des points intérieurs

3.2.2 Exemple

Problème linéaire

- Forme standard : $\min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3$ sous $\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$ $A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$, $b=1$
 $c^T = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$

- Conditions d'ordre 1 du problème linéaire (PL)

$$XS = 0 \Rightarrow \begin{cases} x_1 s_1 = 0 \\ x_2 s_2 = 0 \\ x_3 s_3 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = 0 \text{ ou } s_1 = 0 \\ x_2 = 0 \text{ ou } s_2 = 0 \\ x_3 = 0 \text{ ou } s_3 = 0 \end{cases} \rightarrow 6 \text{ combinaisons possibles}$$

$$A^T \lambda + s = c \Rightarrow \begin{cases} \lambda + s_1 = 1 \\ \lambda + s_2 = 2 \\ \lambda + s_3 = 3 \end{cases} \Rightarrow \begin{cases} s_1 = 1 - \lambda \\ s_2 = 2 - \lambda \\ s_3 = 3 - \lambda \end{cases} \Rightarrow \begin{cases} s_1 = s_2 - 1 \\ s_2 = s_3 - 1 \end{cases}$$

$$S \geq 0 \Rightarrow \begin{cases} s_1 \geq 0 \\ s_2 \geq 1 \\ s_3 \geq 2 \end{cases} \Rightarrow \begin{cases} x_1 = 0 \text{ ou } s_1 = 0 \\ x_2 = 0 \\ x_3 = 0 \end{cases} \rightarrow 2 \text{ combinaisons possibles}$$

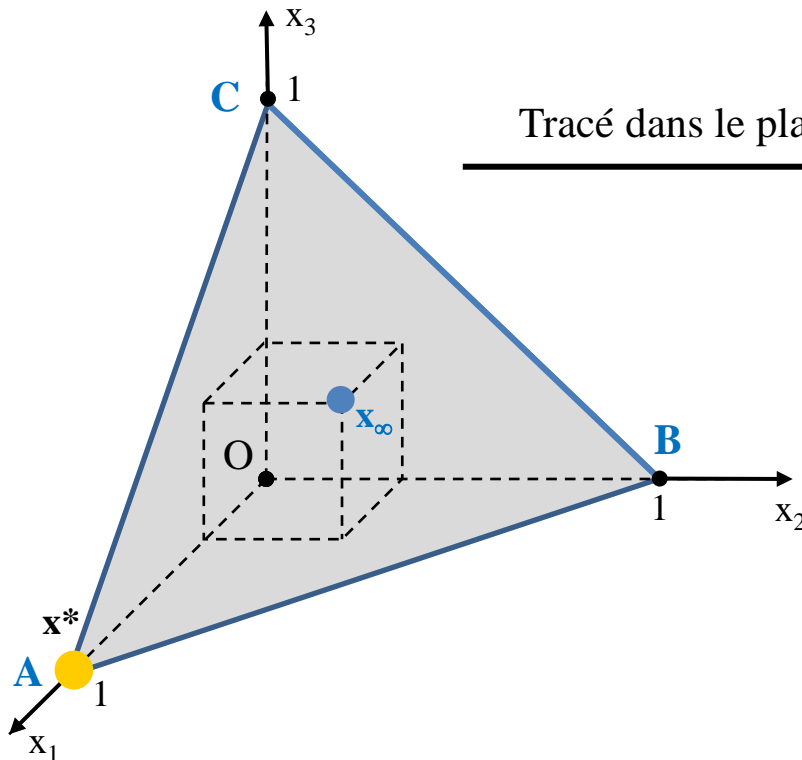
$$Ax - b = 0 \Rightarrow x_1 + x_2 + x_3 = 1 \Rightarrow x_1 = 1$$

- Solution : $\begin{cases} x_1 = 1 \\ x_2 = 0 \\ x_3 = 0 \end{cases}, \begin{cases} s_1 = 0 \\ s_2 = 1 \\ s_3 = 2 \end{cases}, \lambda = 1$

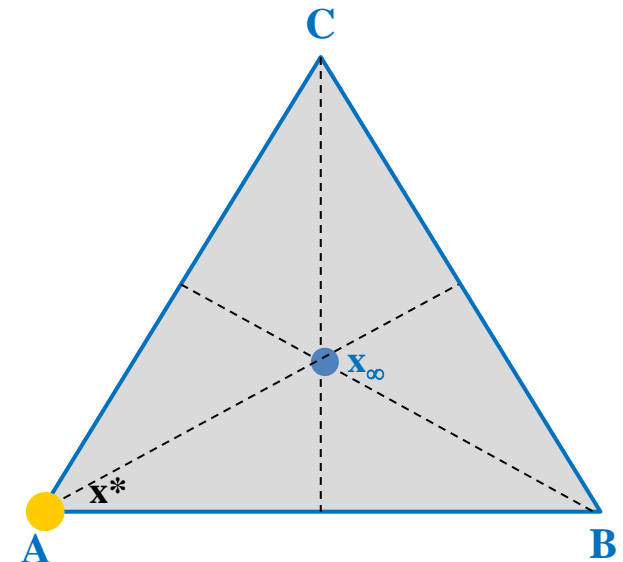
3.2.2 Exemple

Représentation graphique

- Centre analytique du polytope : $P = \left\{ (x_1, x_2, x_3) \in \mathbb{R}^3 / \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases} \right\} \rightarrow x_\infty = \left(\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \right)$
- Solution du problème (PL) : $\min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3$ sous $\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases} \rightarrow x^* = (1 \quad 0 \quad 0)$
 \rightarrow **point A**



Tracé dans le plan (A,B,C)



3.2.2 Exemple

Problème barrière

- Problème linéaire : $\min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3$ sous $\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$ $A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$, $b=1$
 $c^T = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$
- Conditions d'ordre 1 du problème barrière (PB_h)

$$A^T \lambda + s = c \Rightarrow \begin{cases} \lambda + s_1 = 1 \\ \lambda + s_2 = 2 \\ \lambda + s_3 = 3 \end{cases} \Rightarrow \begin{cases} s_1 = 1 - \lambda \\ s_2 = 2 - \lambda \\ s_3 = 3 - \lambda \end{cases}$$

$$XS = he \Rightarrow \begin{cases} x_1 s_1 = h \\ x_2 s_2 = h \\ x_3 s_3 = h \end{cases} \Rightarrow \begin{cases} x_1 = \frac{h}{1 - \lambda} = \frac{h}{\mu - 1} \\ x_2 = \frac{h}{2 - \lambda} = \frac{h}{\mu} \\ x_3 = \frac{h}{3 - \lambda} = \frac{h}{\mu + 1} \end{cases} \text{ avec } \mu = 2 - \lambda$$

$$Ax - b = 0 \Rightarrow x_1 + x_2 + x_3 = 1 \Rightarrow \frac{h}{\mu - 1} + \frac{h}{\mu} + \frac{h}{\mu + 1} = 1 \Rightarrow \mu^3 - 3h\mu^2 - \mu + h = 0$$

- On obtient une équation en $\mu \rightarrow 1, 2$ ou 3 racines

3.2.2 Exemple

Chemin central

- Le point du chemin central (x_h, λ_h, s_h) pour une barrière de hauteur h vérifie

$$\begin{cases} Ax_h - b = 0 \\ A^T \lambda_h + s_h - c = 0 \\ X_h S_h - h e = 0 \end{cases} \text{ avec } \begin{cases} x_h \geq 0 \\ s_h \geq 0 \end{cases}$$

- En résolvant les conditions d'ordre 1 du problème barrière, on obtient

$$\begin{cases} \lambda_h = 2 - \mu & \text{avec } \mu^3 - 3h\mu^2 - \mu + h = 0 \\ s_h = \begin{pmatrix} 1 - \lambda_h & 2 - \lambda_h & 3 - \lambda_h \end{pmatrix} \\ x_h = h \begin{pmatrix} \frac{1}{1 - \lambda_h} & \frac{1}{2 - \lambda_h} & \frac{1}{3 - \lambda_h} \end{pmatrix} \end{cases}$$

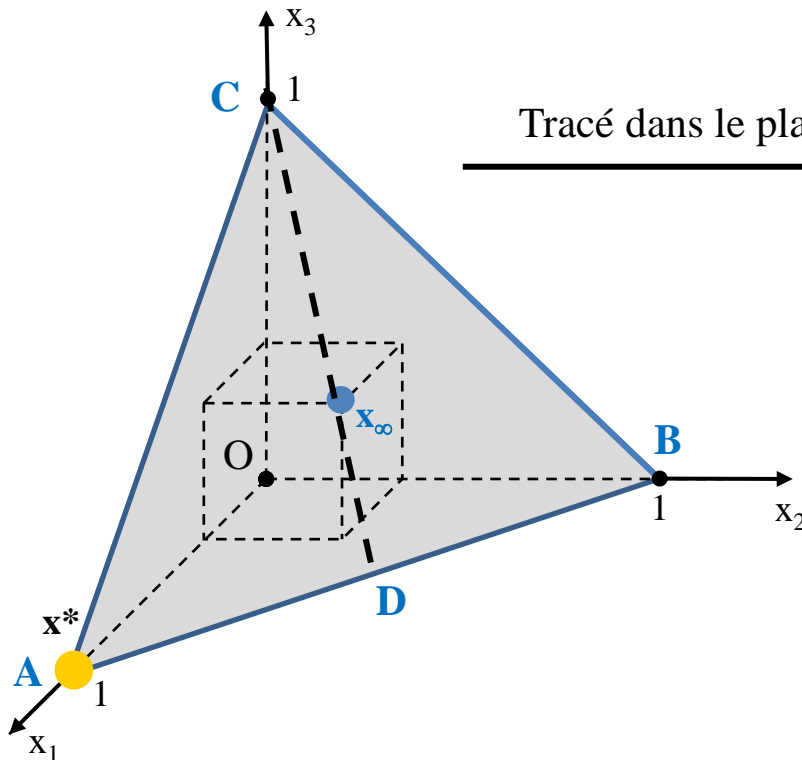
- Il faut vérifier $\begin{cases} x_h \geq 0 \\ s_h \geq 0 \end{cases} \rightarrow$ choix parmi les racines possibles pour μ
- On résout le problème pour des valeurs décroissantes de la hauteur de barrière h .

3.2.2 Exemple

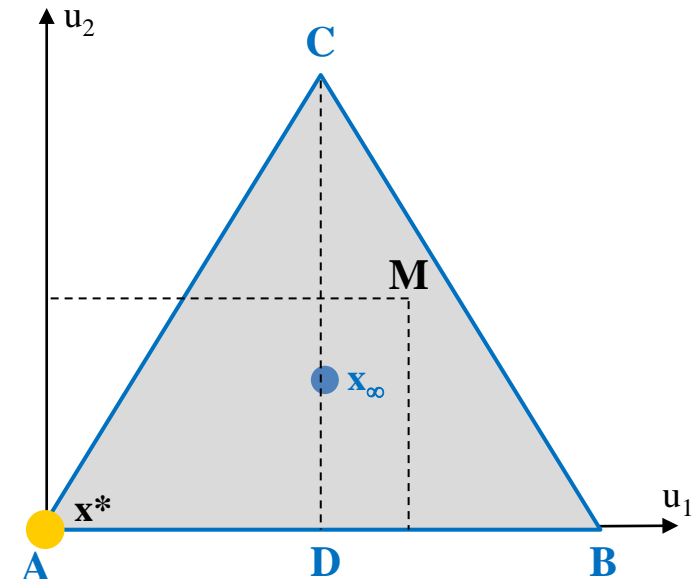
Représentation graphique

- Système de coordonnées (u_1, u_2) dans le plan (A,B,C)

$$\overrightarrow{AM} = u_1 \overrightarrow{AB} + u_2 \overrightarrow{DC} \Leftrightarrow \begin{pmatrix} x_1 - 1 \\ x_2 \\ x_3 \end{pmatrix} = u_1 \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} + u_2 \begin{pmatrix} -1/2 \\ -1/2 \\ 1 \end{pmatrix} \Rightarrow \begin{cases} u_1 = x_2 + \frac{1}{2}x_3 \\ u_2 = x_3 \end{cases}$$



Tracé dans le plan (A,B,C)



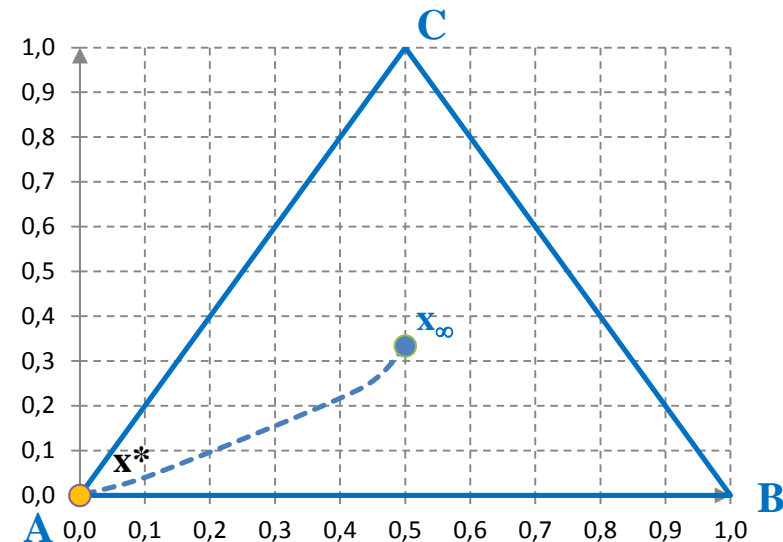
3.2.2 Exemple

Chemin central

- Centre analytique du polytope ($h \rightarrow \infty$) : $x_\infty = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$
- Points du chemin central pour une barrière de hauteur h de 10000 à 0.

h	x1	x2	x3	s1	s2	s3	λ
10000	0,33335	0,33334	0,33333	29998,5	29999,5	30000,5	-29997,5
1000	0,33342	0,33331	0,33320	2999,2	3000,2	3001,2	-2998,2
100	0,33444	0,33332	0,33222	299,0	300,0	301,0	-298,0
10	0,34457	0,33309	0,32236	29,0	30,0	31,0	-28,0
1	0,45162	0,31112	0,23729	2,2142	3,2142	4,2142	-1,2142
0,1000	0,86308	0,08962	0,04726	0,1159	1,1159	2,1159	0,8841
0,0100	0,98507	0,00990	0,00497	0,0102	1,0102	2,0102	0,9898
0,0010	0,99863	0,00100	0,00050	0,0010	1,0010	2,0010	0,9990
0,0001	0,99970	0,00010	0,00005	0,0001	1,0001	2,0001	0,9999
0,0000	1,00000	0,00000	0,00000	0,0000	1,0000	2,0000	1,0000

Tracé dans le plan (A,B,C)



- Solution du problème linéaire ($h \rightarrow 0$) : $x^* = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$

3.2.2 Déplacement

Méthode de Newton

- On cherche à résoudre
$$\begin{cases} Ax - b = 0 \\ A^T \lambda + s - c = 0 \\ XS - he = 0 \end{cases} \text{ pour } h \text{ fixé avec } \begin{cases} x \geq 0 \\ s \geq 0 \end{cases}$$

- On définit la fonction :
$$F(x, \lambda, s) = \begin{pmatrix} Ax - b \\ A^T \lambda + s - c \\ XS - he \end{pmatrix} \Rightarrow \nabla F(x, \lambda, s)^T = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix}$$

- On applique la méthode de Newton au système d'équations : $F(x, \lambda, s) = 0$

- Itération de Newton :
$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = -F(x, \lambda, s)$$

$$\text{avec } F(x, \lambda, s) = \begin{pmatrix} 0 \\ 0 \\ XSe - he \end{pmatrix} \text{ si } (x, \lambda, s) \text{ est intérieur : } \begin{cases} Ax - b = 0 \\ A^T \lambda + s - c = 0 \end{cases}$$

→ direction de déplacement (d_x, d_λ, d_s)

3.2.2 Déplacement

Méthode de Newton

- On utilise la solution de Newton comme direction de recherche.

$$\begin{pmatrix} \mathbf{x}_h \\ \lambda_h \\ \mathbf{s}_h \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \lambda \\ \mathbf{s} \end{pmatrix} + \alpha \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_\lambda \\ \mathbf{d}_s \end{pmatrix}, 0 < \alpha \leq 1$$

- Le pas de déplacement α est choisi pour rester dans l'ensemble des points intérieurs X_{int} .

$$\begin{cases} \mathbf{x}_h \geq 0 \\ \mathbf{s}_h \geq 0 \end{cases}$$

Convergence

Pour que l'algorithme converge vers la solution du problème linéaire (PL), il faut :

- régler le pas α pour ne pas s'approcher trop rapidement du bord de l'ensemble admissible
- abaisser progressivement la hauteur de la barrière h jusqu'à 0

La solution du problème linéaire PL est obtenue lorsque : $\mathbf{XS} = 0 \Leftrightarrow \mathbf{x}_i \mathbf{s}_i = 0, i = 1, \dots, n$

3.2.2 Mesure de dualité

Mesure de dualité

- La **mesure de dualité** est définie par : $v = \frac{1}{n} x^T s = \frac{1}{n} \sum_{i=1}^n x_i s_i$
 → distance moyenne à la condition d'optimalité $XS = 0 \Leftrightarrow x_i s_i = 0, i = 1, \dots, n$
- La hauteur de barrière est réglée à partir de la mesure de dualité : $h = \sigma v$
- σ est le **paramètre de centrage**
 Le paramètre de centrage permet de corriger la direction de déplacement
 $\sigma = 0$: pas de barrière
 → La direction donnée par l'itération de Newton vise à résoudre les conditions d'optimalité du problème initial (PL).
 → Peu robuste loin de la solution, blocage au bord du polytope
 $\sigma = 1$: barrière $h=v$
 → La direction donnée par l'itération de Newton vise à revenir sur le point du chemin central correspondant à $h=v$.
 → Permet de rester à l'intérieur du polytope

3.2.2 Exemple

Problème linéaire

- Problème linéaire : $\min_{x_1, x_2, x_3} x_1 + 2x_2 + 3x_3$ sous $\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1, x_2, x_3 \geq 0 \end{cases}$ $A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$, $b=1$
 $c^T = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$

- On cherche à résoudre : $F(x, \lambda, s) = \begin{pmatrix} Ax - b \\ A^T \lambda + s - c \\ XS - he \end{pmatrix} = 0$ pour h fixé avec $\begin{cases} x \geq 0 \\ s \geq 0 \end{cases}$
 → méthode de Newton à partir d'un point intérieur initial

Point initial

- On choisit un point intérieur initial $(x, \lambda, s) \in X_{\text{int}} \Rightarrow \begin{cases} Ax - b = 0 \\ A^T \lambda + s - c = 0 \end{cases}$ et $\begin{cases} x > 0 \\ s > 0 \end{cases}$

On peut prendre $\begin{cases} (x_1, x_2, x_3) > 0 \text{ telque } x_1 + x_2 + x_3 = 1 \\ \lambda = 0 \\ s = c \Rightarrow s^T = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} > 0 \end{cases} \Rightarrow v = \frac{x^T s}{n} = \frac{x_1 + 2x_2 + 3x_3}{3}$

- La hauteur de barrière h est réglée par le paramètre de centrage σ : $h = \sigma v = \frac{\sigma}{n} x^T s$

3.2.2 Exemple

Direction de déplacement

La direction de déplacement (d_x, d_λ, d_s) à partir du point initial (x, λ, s) est obtenue en résolvant les équations de Newton.

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = -F(x, \lambda, s) = \begin{pmatrix} 0 \\ 0 \\ -XS + he \end{pmatrix} \quad \text{car} \quad \begin{cases} Ax - b = 0 \\ A^T \lambda + s - c = 0 \end{cases}$$

$$\Leftrightarrow \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ s_1 & 0 & 0 & 0 & x_1 & 0 & 0 \\ 0 & s_2 & 0 & 0 & 0 & x_2 & 0 \\ 0 & 0 & s_3 & 0 & 0 & 0 & x_3 \end{pmatrix} \begin{pmatrix} d_{x1} \\ d_{x2} \\ d_{x3} \\ d_\lambda \\ d_{s1} \\ d_{s2} \\ d_{s3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -x_1 s_1 + h \\ -x_2 s_2 + h \\ -x_3 s_3 + h \end{pmatrix} \quad \text{avec} \quad h = \frac{\sigma}{n} (x_1 s_1 + x_2 s_2 + x_3 s_3)$$

3.2.2 Exemple

Direction de déplacement

$$\begin{cases} d_{x1} + d_{x2} + d_{x3} = 0 \\ d_{s1} + d_{\lambda} = 0 \\ d_{s2} + d_{\lambda} = 0 \\ d_{s3} + d_{\lambda} = 0 \\ s_1 d_{x1} + x_1 d_{s1} = h - x_1 s_1 \\ s_2 d_{x2} + x_2 d_{s2} = h - x_2 s_2 \\ s_3 d_{x3} + x_3 d_{s3} = h - x_3 s_3 \end{cases} \Leftrightarrow \begin{cases} d_{\lambda} \left(\frac{x_1}{s_1} + \frac{x_2}{s_2} + \frac{x_3}{s_3} \right) = x_1 + x_2 + x_3 - h \left(\frac{1}{s_1} + \frac{1}{s_2} + \frac{1}{s_3} \right) \\ d_{s1} = d_{s2} = d_{s3} = -d_{\lambda} \\ d_{x1} = \frac{h + x_1 d_{\lambda}}{s_1} - x_1 \\ d_{x2} = \frac{h + x_2 d_{\lambda}}{s_2} - x_2 \\ d_{x3} = \frac{h + x_3 d_{\lambda}}{s_3} - x_3 \end{cases} \quad \text{avec} \quad \begin{cases} s_1 = 1 \\ s_2 = 2 \\ s_3 = 3 \end{cases} \quad \text{et} \quad \begin{cases} v = \frac{x_1 + 2x_2 + 3x_3}{3} \\ h = \sigma v \end{cases}$$

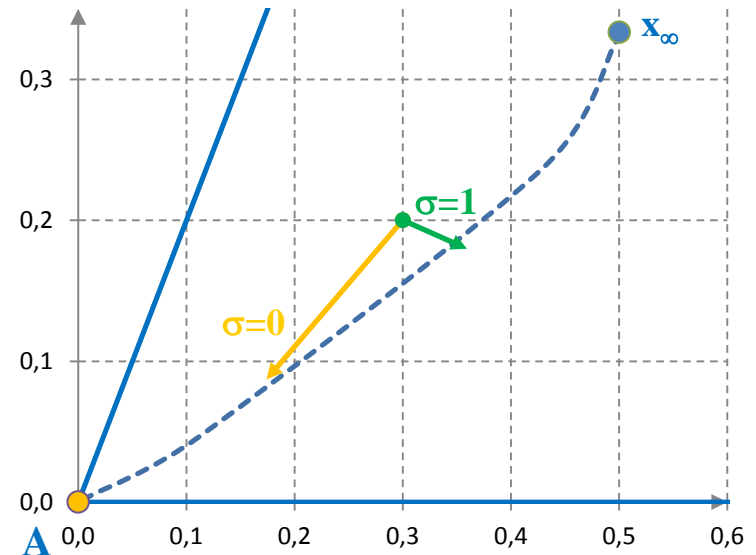
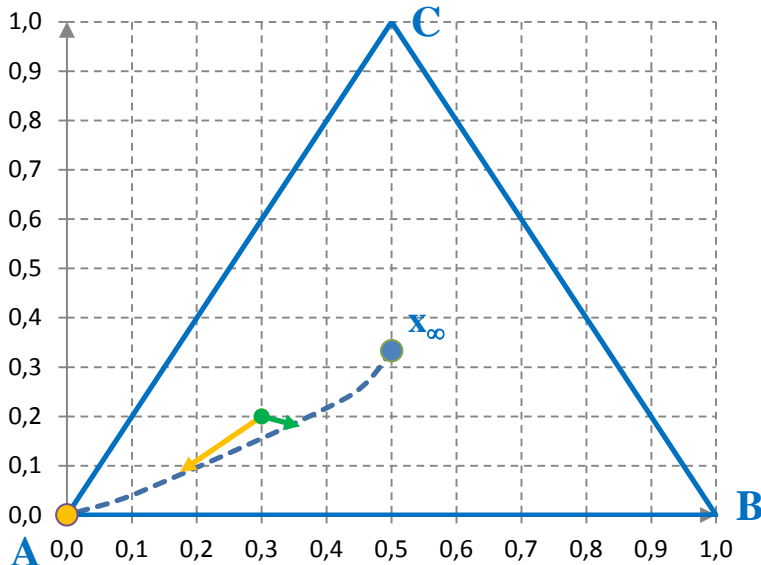
- Le point initial (x_1, x_2, x_3) doit vérifier $\begin{cases} x_1, x_2, x_3 > 0 \\ x_1 + x_2 + x_3 = 1 \end{cases}$
- Illustrations pour 3 points initiaux** : $x = (0.6, 0.2, 0.2)$
 $x = (0.2, 0.6, 0.2)$
 $x = (0.2, 0.2, 0.6)$
 et pour **2 valeurs de σ** : $\sigma = 0 \rightarrow$ vers la solution du problème initial (Newton)
 $\sigma = 1 \rightarrow$ vers le chemin central (x_h, λ_h, s_h)

3.2.2 Exemple

Illustration 1 : $x = (0.6, 0.2, 0.2)$

	v	x1	x2	x3	s1	s2	s3	λ
	0,53333	0,6	0,2	0,2	1,0	2,0	3,0	0,0
σ	h	dx1	dx2	dx3	ds1	ds2	ds3	d λ
0	0,00000	0,18261	-0,06957	-0,11304	-1,30435	-1,30435	-1,30435	1,30435
1	0,53333	-0,04928	0,06957	-0,02029	-0,02899	-0,02899	-0,02899	0,02899
Chemin central	0,53333	0,54971	0,27070	0,17956	0,97020	1,97020	2,97020	0,02980

Tracé dans le plan (A,B,C) $\rightarrow \begin{cases} u_1 = x_2 + x_3 / 2 \\ u_2 = x_3 \end{cases}$

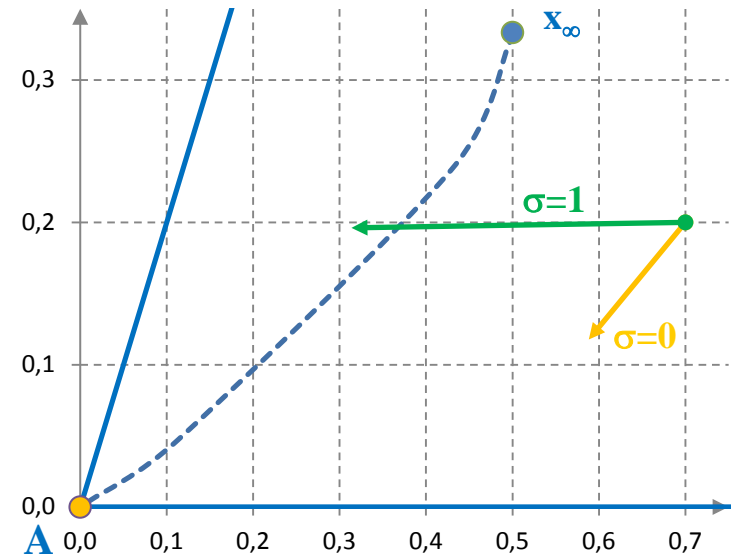
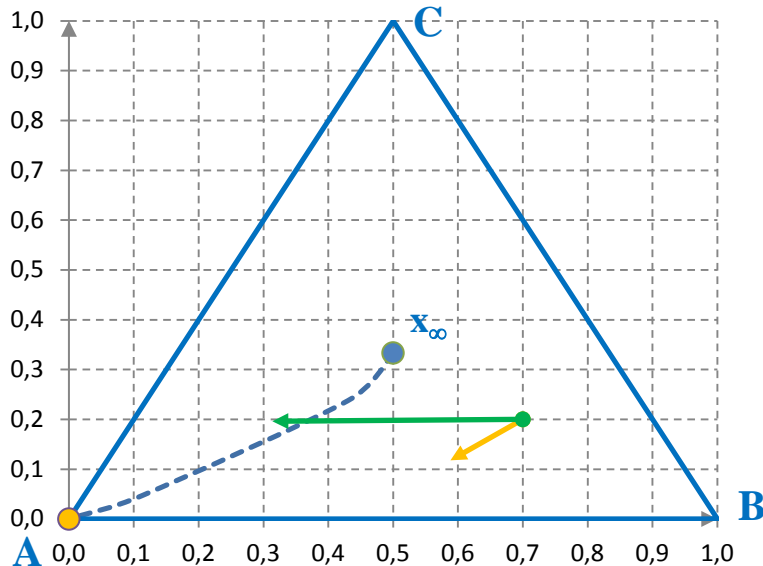


3.2.2 Exemple

Illustration 2 : $x = (0.2, 0.6, 0.2)$

	v	x1	x2	x3	s1	s2	s3	λ
	0,66667	0,2	0,6	0,2	1,0	2,0	3,0	0,0
σ	h	dx1	dx2	dx3	ds1	ds2	ds3	d λ
0	0,00000	0,15294	-0,07059	-0,08235	-1,76471	-1,76471	-1,76471	1,76471
1	0,66667	0,38824	-0,38431	-0,00392	0,39216	0,39216	0,39216	-0,39216
Chemin central	0,66667	0,50965	0,28884	0,20153	1,30808	2,30808	3,30808	-0,30808

Tracé dans le plan (A,B,C) $\rightarrow \begin{cases} u_1 = x_2 + x_3 / 2 \\ u_2 = x_3 \end{cases}$

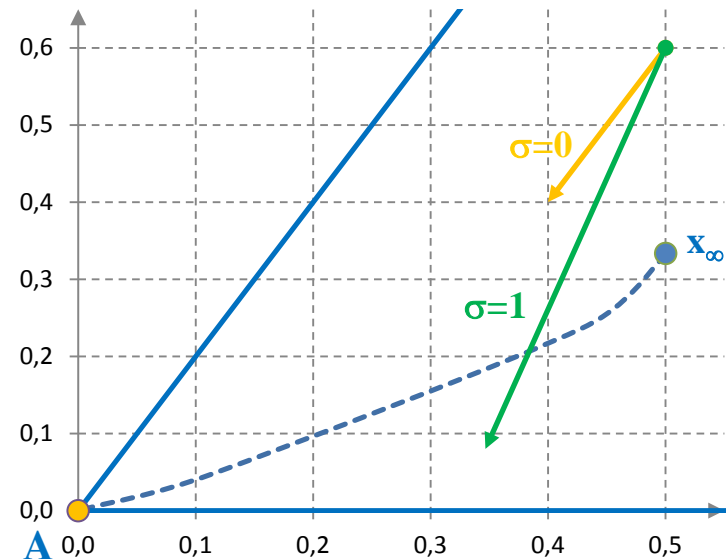
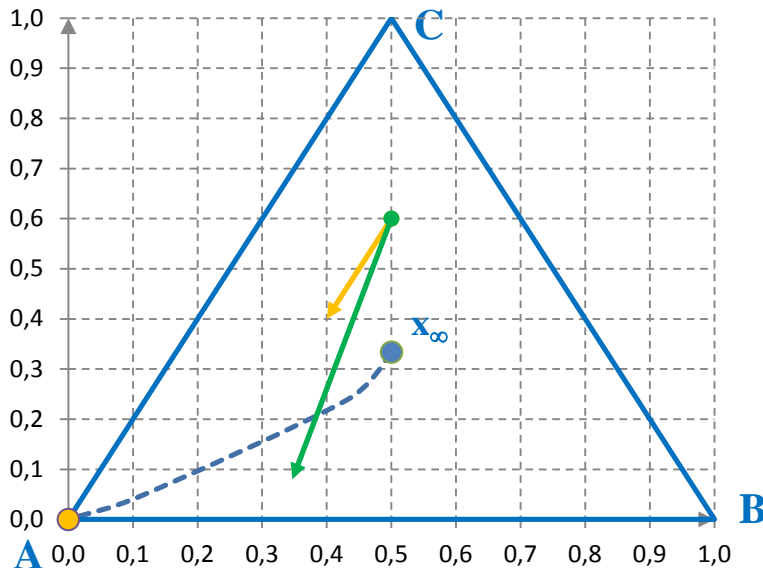


3.2.2 Exemple

Illustration 3 : $x = (0.2, 0.2, 0.6)$

	v	x1	x2	x3	s1	s2	s3	λ
	0,80000	0,2	0,2	0,6	1,0	2,0	3,0	0,0
σ	h	dx1	dx2	dx3	ds1	ds2	ds3	d λ
0	0,00000	0,20000	0,00000	-0,20000	-2,00000	-2,00000	-2,00000	2,00000
1	0,80000	0,41333	0,10667	-0,52000	0,93333	0,93333	0,93333	-0,93333
Chemin central	0,80000	0,48130	0,30051	0,21845	1,66217	2,66217	3,66217	-0,66217

Tracé dans le plan (A,B,C) $\rightarrow \begin{cases} u_1 = x_2 + x_3 / 2 \\ u_2 = x_3 \end{cases}$



3.2.2 Mesure de dualité

Distance au chemin central

- Pour une hauteur de barrière h , le point du chemin central est tel que tous les produits $x_i s_i$ sont égaux à h : $XS = he \Leftrightarrow x_i s_i = h, i = 1, \dots, n$
- Au point courant la moyenne des produits $x_i s_i$ est la mesure de dualité v : $v = \frac{1}{n} x^T s = \frac{1}{n} \sum_{i=1}^n x_i s_i$
- On quantifie la **distance δ du point courant au chemin central** par la moyenne des écarts $(x_i s_i - v)$

$$\delta = \frac{1}{v} \left\| \begin{pmatrix} x_1 s_1 \\ \vdots \\ x_n s_n \end{pmatrix} - \begin{pmatrix} v \\ \vdots \\ v \end{pmatrix} \right\| = \frac{1}{v} \|XS e - v e\|$$

- Pour que l'algorithme de point intérieur converge, il faut que les produits $x_i s_i$ tendent simultanément vers 0, et éviter que certains s'annulent prématurément.
- On impose de suivre approximativement le chemin central en interdisant de trop s'en écarter
 → contrainte de distance maximale au chemin central : $\|\delta\| \leq \delta_m$
 → définition du **voisinage du chemin central**

3.2.2 Voisinage

Voisinage du chemin central

Le voisinage du chemin central est défini par une borne δ_m sur la distance δ : $\|\delta\| \leq \delta_m$

- **Voisinage restreint avec la norme 2**, noté $V_2(\delta_m)$

$$V_2(\delta_m) = \left\{ (x, \lambda, s) \in X_{\text{int}} / \frac{1}{v} \|XSe - ve\|_2 \leq \delta_m \right\} \text{ avec } 0 \leq \delta_m < 1$$

- **Voisinage large avec la norme ∞** , noté $V_\infty(\delta_m)$

$$V_\infty(\delta_m) = \left\{ (x, \lambda, s) \in X_{\text{int}} / \frac{1}{v} \|XSe - ve\|_\infty \leq \delta_m \right\} \text{ avec } 0 \leq \delta_m < 1$$

$$\begin{aligned} \frac{1}{v} \|XSe - ve\|_\infty \leq \delta_m &\Leftrightarrow |x_i s_i - v| \leq v \delta_m, i = 1, \dots, n \\ &\Leftrightarrow v(1 - \delta_m) \leq x_i s_i \leq v(1 + \delta_m), i = 1, \dots, n \end{aligned}$$

- On se contente de la borne inférieure, qui empêche les produits $x_i s_i$ de converger prématurément vers 0. En remplaçant δ_m par $1 - \delta_m$

$$V_{-\infty}(\delta_m) = \left\{ (x, \lambda, s) \in X_{\text{int}} / x_i s_i \geq v \delta_m, i = 1, \dots, n \right\} \text{ avec } 0 \leq \delta_m < 1$$

3.2.3 Algorithmes

- ☐ Suivi du chemin central
- ☐ Algorithme à pas restreint
- ☐ Algorithme à pas long
- ☐ Algorithme de prédiction-correction
- ☐ Exemple

3.2.3 Algorithmes

Suivi du chemin central

On peut envisager 3 algorithmes de suivi du chemin central

- Algorithme à pas restreint
- Algorithme à pas long
- Algorithme de prédiction-correction

Principes

- Les 3 algorithmes sont basés sur l'itération de Newton

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ X_k S_k e - h e \end{pmatrix} \quad \text{avec} \quad \begin{cases} h = \sigma v_k \\ v_k = \frac{1}{n} x_k^T S_k \end{cases}$$

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ s_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \\ s_k \end{pmatrix} + \alpha \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix}, \quad 0 < \alpha \leq 1 \quad \text{avec } \alpha \text{ choisi tel que } \begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ s_{k+1} \end{pmatrix} \in V_2(\delta_m) \text{ ou } V_{-\infty}(\delta_m)$$

- Les différences résident dans la stratégie de réglage du paramètre de centrage σ et du pas α .

3.2.3 Algorithme à pas restreint

Algorithme à pas restreint

- On applique systématiquement l'itération de Newton avec $\alpha = 1$.

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \lambda_{k+1} \\ \mathbf{s}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k \\ \lambda_k \\ \mathbf{s}_k \end{pmatrix} + \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_\lambda \\ \mathbf{d}_s \end{pmatrix}$$

- On règle le paramètre de centrage σ pour rester dans le voisinage restreint du chemin central.

$$\begin{pmatrix} \mathbf{A} & 0 & 0 \\ 0 & \mathbf{A}^T & \mathbf{I} \\ \mathbf{S}_k & 0 & \mathbf{X}_k \end{pmatrix} \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_\lambda \\ \mathbf{d}_s \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ \mathbf{X}_k \mathbf{S}_k \mathbf{e} - \sigma \mathbf{v}_k \mathbf{e} \end{pmatrix} \quad \text{avec } \sigma \text{ tel que : } \begin{pmatrix} \mathbf{x}_{k+1} \\ \lambda_{k+1} \\ \mathbf{s}_{k+1} \end{pmatrix} \in \mathbf{V}_2(\delta_m)$$

Réglages

- $\delta_m = 0.4$ → largeur du voisinage
- $\sigma = 1 - \frac{\delta_m}{\sqrt{n}}$ → garantit que l'itération de Newton reste dans le voisinage restreint

3.2.3 Algorithme à pas restreint

Réglage du paramètre de centrage

Le nouveau point doit rester dans le voisinage restreint :
$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \lambda_{k+1} \\ \mathbf{s}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k \\ \lambda_k \\ \mathbf{s}_k \end{pmatrix} + \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_\lambda \\ \mathbf{d}_s \end{pmatrix} \in V_2(\delta_m)$$

$$\Rightarrow \frac{1}{\mathbf{v}_k} \|\mathbf{X}_{k+1} \mathbf{S}_{k+1} \mathbf{e} - \mathbf{v}_k \mathbf{e}\|_2 \leq \delta_m \quad \text{avec} \quad \begin{cases} \mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{D}_x, & \mathbf{D}_x = \text{diag}(\mathbf{d}_x), \mathbf{d}_x = \mathbf{D}_x \mathbf{e} \\ \mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{D}_s, & \mathbf{D}_s = \text{diag}(\mathbf{d}_s), \mathbf{d}_s = \mathbf{D}_s \mathbf{e} \\ \mathbf{S}_k \mathbf{d}_x + \mathbf{X}_k \mathbf{d}_s = -\mathbf{X}_k \mathbf{S}_k \mathbf{e} + \sigma \mathbf{v}_k \mathbf{e} \end{cases}$$

$$\Rightarrow \frac{1}{\mathbf{v}_k} \|(\mathbf{X}_k + \mathbf{D}_x)(\mathbf{S}_k + \mathbf{D}_s) \mathbf{e} - \mathbf{v}_k \mathbf{e}\|_2 \leq \delta_m$$

$$\Rightarrow \frac{1}{\mathbf{v}_k} \|\mathbf{X}_k \mathbf{S}_k \mathbf{e} + \mathbf{X}_k \mathbf{d}_s + \mathbf{S}_k \mathbf{d}_x - \mathbf{v}_k \mathbf{e}\|_2 \leq \delta_m \quad \text{à l'ordre 1 en } \mathbf{d}_x, \mathbf{d}_s$$

$$\Rightarrow \frac{1}{\mathbf{v}_k} \|\sigma \mathbf{v}_k \mathbf{e} - \mathbf{v}_k \mathbf{e}\|_2 \leq \delta_m$$

$$\Rightarrow |\sigma - 1| \|\mathbf{e}\|_2 \leq \delta_m \quad \text{avec} \quad \|\mathbf{e}\|_2 = \sqrt{n} \quad \Rightarrow \quad \boxed{\sigma \geq 1 - \frac{\delta_m}{\sqrt{n}}}$$

3.2.3 Algorithme à pas long

Algorithme à pas long

- On fixe le paramètre de centrage σ

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ X_k S_k e - \sigma v_k e \end{pmatrix}$$

- On règle le pas α pour rester dans le voisinage large du chemin central.

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ s_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \\ s_k \end{pmatrix} + \alpha \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix}, 0 < \alpha \leq 1 \quad \text{avec } \alpha \text{ tel que : } \begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ s_{k+1} \end{pmatrix} \in V_{-\infty}(\delta_m)$$

Réglages

- $\delta_m = 0.001 \rightarrow$ largeur du voisinage
- $\sigma = 0.1$
- Initialisation avec $\alpha=1$

Division de α par 2 tant que le nouveau point n'est pas dans le voisinage large

3.2.3 Algorithme de prédiction-correction

Algorithme de prédiction-correction

Etape de prédiction

- On prédit la **direction de l'optimum** avec un paramètre de centrage $\sigma=0$ (pas de barrière).

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \tilde{d}_x \\ \tilde{d}_\lambda \\ \tilde{d}_s \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ X_k S_k e \end{pmatrix}$$

- On règle le pas α pour rester dans le voisinage restreint du chemin central.

$$\begin{pmatrix} \tilde{x}_{k+1} \\ \tilde{\lambda}_{k+1} \\ \tilde{s}_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \\ s_k \end{pmatrix} + \alpha \begin{pmatrix} \tilde{d}_x \\ \tilde{d}_\lambda \\ \tilde{d}_s \end{pmatrix}, 0 < \alpha \leq 1 \quad \text{avec } \alpha \text{ tel que : } \begin{pmatrix} \tilde{x}_{k+1} \\ \tilde{\lambda}_{k+1} \\ \tilde{s}_{k+1} \end{pmatrix} \in V_2(\delta_{\text{mpred}})$$

Réglages

- $\delta_{\text{mpred}} = 0.5 \rightarrow$ largeur du voisinage
- Initialisation avec $\alpha=1$
 Division de α par 2 tant que le nouveau point n'est pas dans le voisinage restreint

3.2.3 Algorithme de prédiction-correction

Algorithme de prédiction-correction

Etape de correction

- On calcule la direction du chemin central avec un paramètre de centrage $\sigma=1$

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ X_k S_k e - v_k e \end{pmatrix}$$

- On applique un **recentrage** avec un pas $\alpha=1$ pour revenir vers le chemin central.

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ s_{k+1} \end{pmatrix} = \begin{pmatrix} \tilde{x}_{k+1} \\ \tilde{\lambda}_{k+1} \\ \tilde{s}_{k+1} \end{pmatrix} + \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix}$$

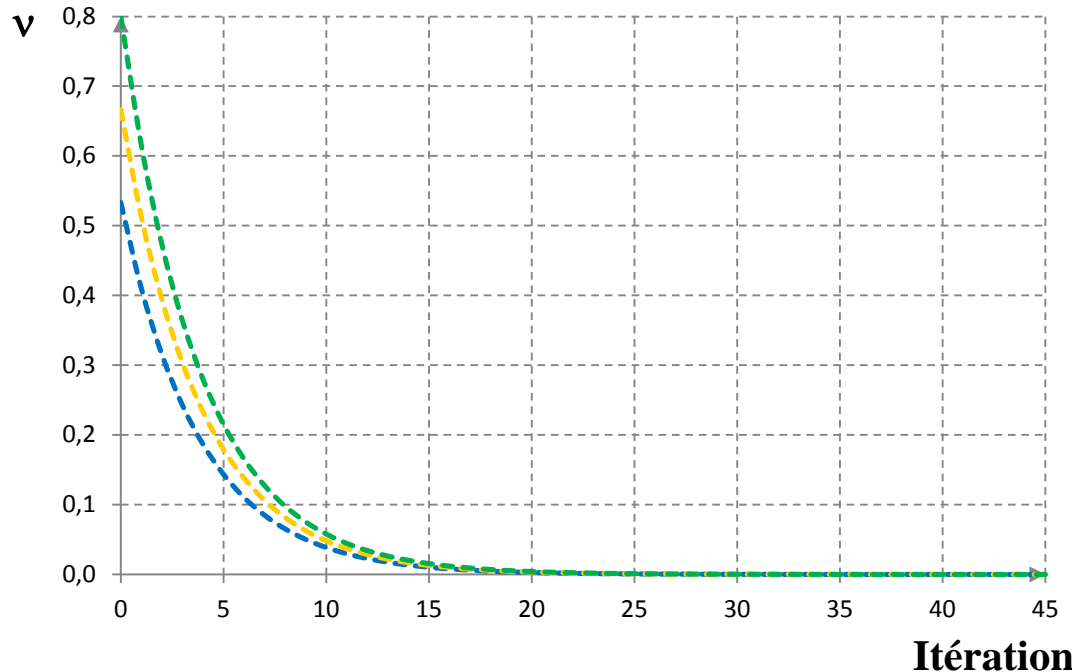
Mise en œuvre pratique

- Différentes stratégies possibles → contrôle de la distance au chemin central
 → réglages (σ, δ, α) à adapter au cours des itérations
- Choix du point initial → suffisamment loin des bords ($x^T s \gg 0$) sinon blocage
- Extension à des problèmes non linéaires

3.2.3 Exemple

Algorithme à pas restreint

- Point initial : $x = (0.6, 0.2, 0.2)$, $\lambda = 0$, $s = (1, 2, 3)$
 $x = (0.2, 0.6, 0.2)$
 $x = (0.2, 0.2, 0.6)$
- Paramètre de centrage : $\sigma = 1 - \frac{\delta_m}{\sqrt{n}}$ avec $\delta_m = 0,4$
 $\rightarrow (x, \lambda, s) \in V_2(\delta_m)$

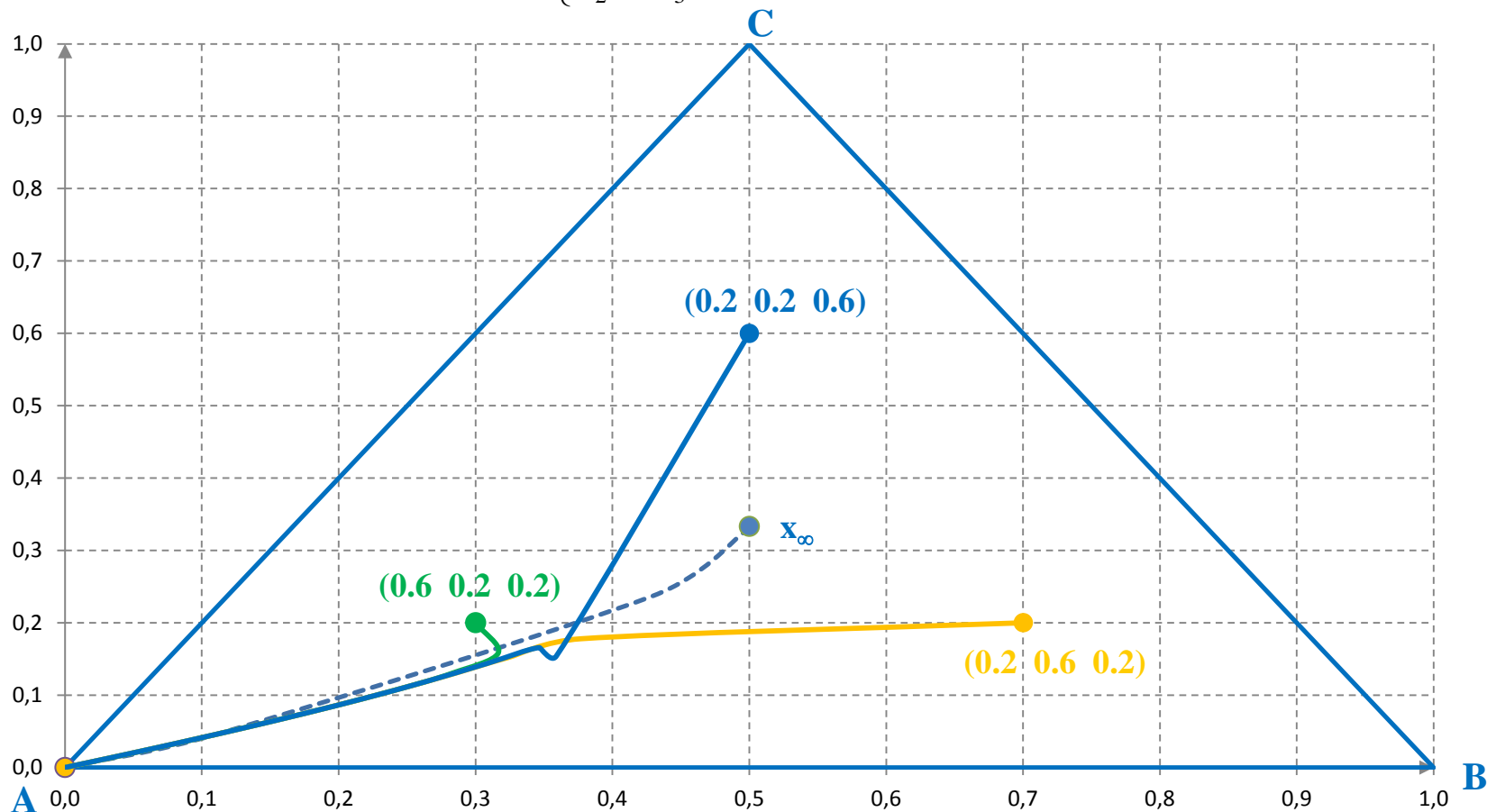


x1	0,6	0,2	0,2
x2	0,2	0,6	0,2
x3	0,2	0,2	0,6
Iteration	v	v	v
0	0,53333	0,66667	0,80000
1	0,41017	0,51271	0,61525
2	0,31544	0,39430	0,47316
3	0,24259	0,30324	0,36389
4	0,18657	0,23321	0,27985
5	0,14348	0,17935	0,21522
6	0,11035	0,13793	0,16552
7	0,08486	0,10608	0,12729
8	0,06526	0,08158	0,09790
9	0,05019	0,06274	0,07529
10	0,03860	0,04825	0,05790
20	0,00279	0,00349	0,00419
30	0,00020	0,00025	0,00030
40	0,00001	0,00002	0,00002
45	0,00000	0,00000	0,00001

3.2.3 Exemple

Algorithme à pas restreint

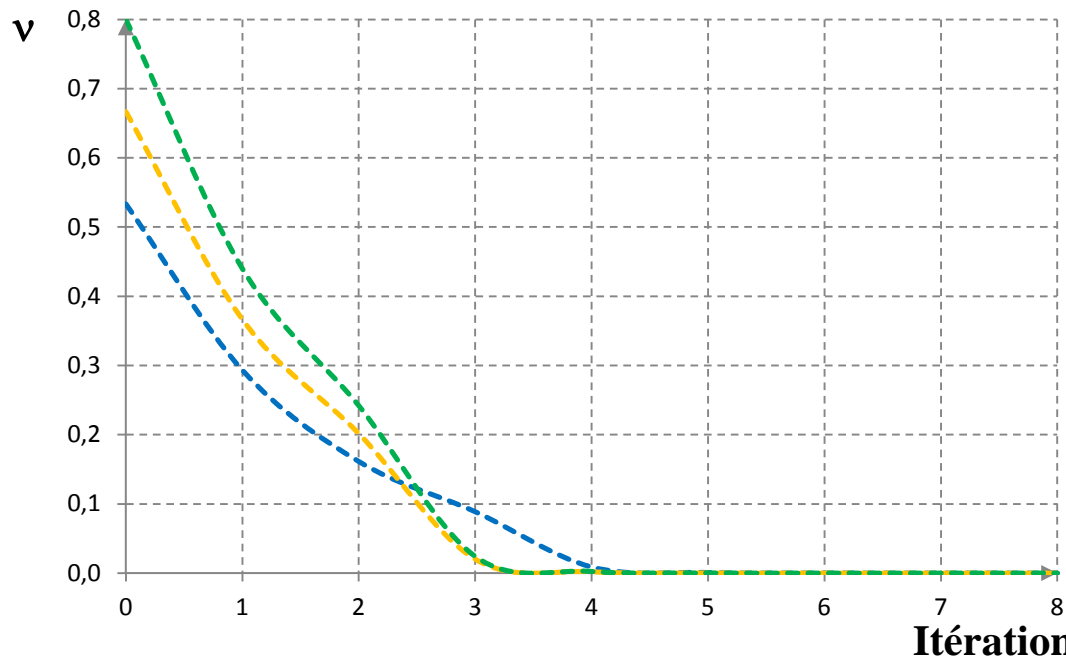
- Tracé dans le plan (A,B,C) $\rightarrow \begin{cases} u_1 = x_2 + x_3 / 2 \\ u_2 = x_3 \end{cases}$



3.2.3 Exemple

Algorithme à pas long

- Point initial : $x = (0.6, 0.2, 0.2)$, $\lambda = 0$, $s = (1, 2, 3)$
 $x = (0.2, 0.6, 0.2)$
 $x = (0.2, 0.2, 0.6)$
- Paramètre de centrage : $\sigma = 0.1$ avec $\delta_m = 0.4$
- Pas : $\alpha=1 \rightarrow \alpha/2$ tant que $(x + \alpha d_x)_i (s + \alpha d_s)_i < v \delta_m$
 $\rightarrow (x, \lambda, s) \in V_{-\infty}(\delta_m)$

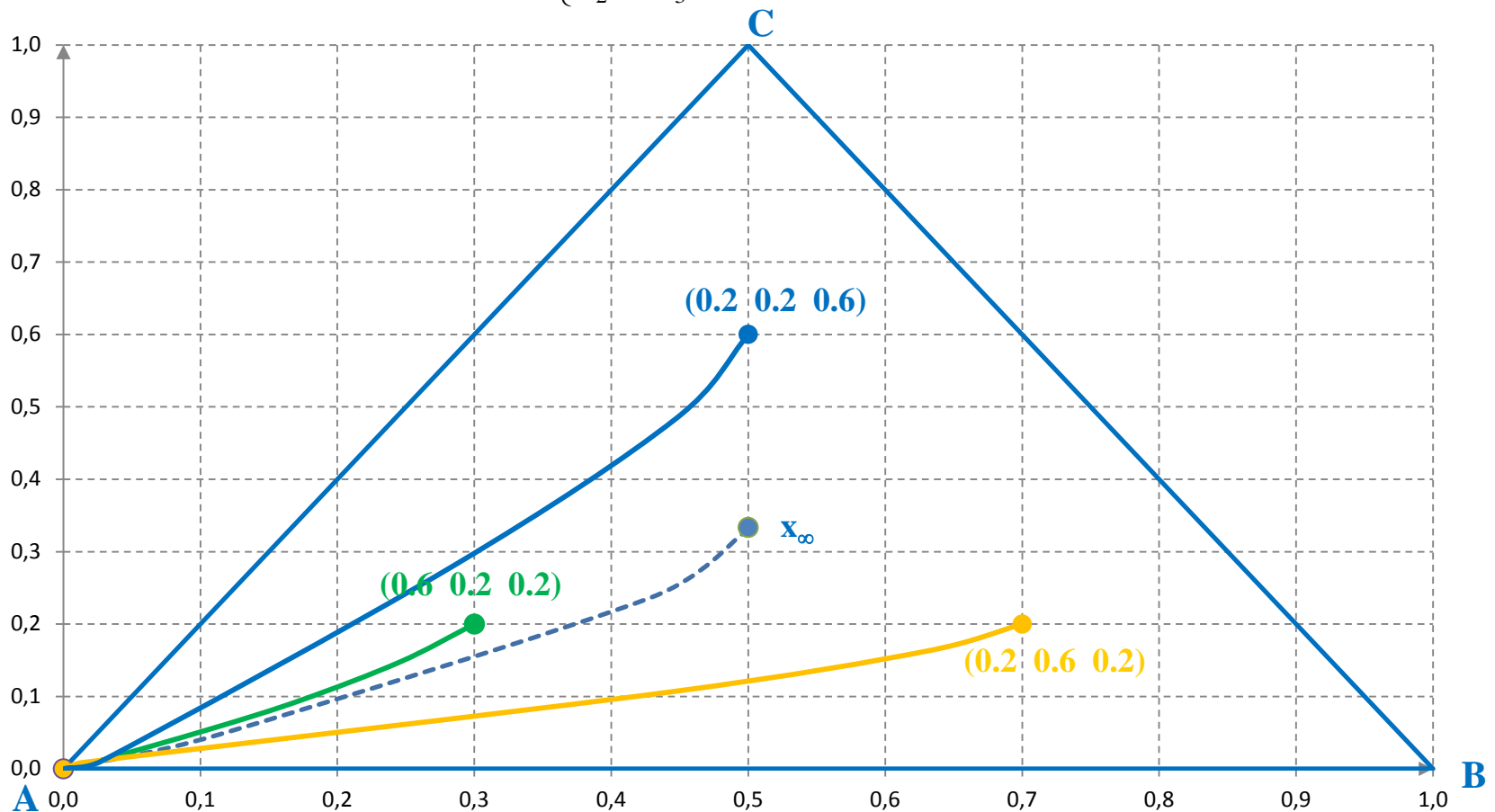


x1	0,6	0,2	0,2
x2	0,2	0,6	0,2
x3	0,2	0,2	0,6
Iteration	v	v	v
0	0,53333	0,66667	0,80000
1	0,29333	0,36667	0,44000
2	0,16133	0,20167	0,24200
3	0,08873	0,02017	0,02420
4	0,00887	0,00202	0,00242
5	0,00089	0,00020	0,00024
6	0,00009	0,00002	0,00002
7	0,00001	0,00000	0,00000
8	0,00000	0,00000	0,00000

3.2.3 Exemple

Algorithme à pas long

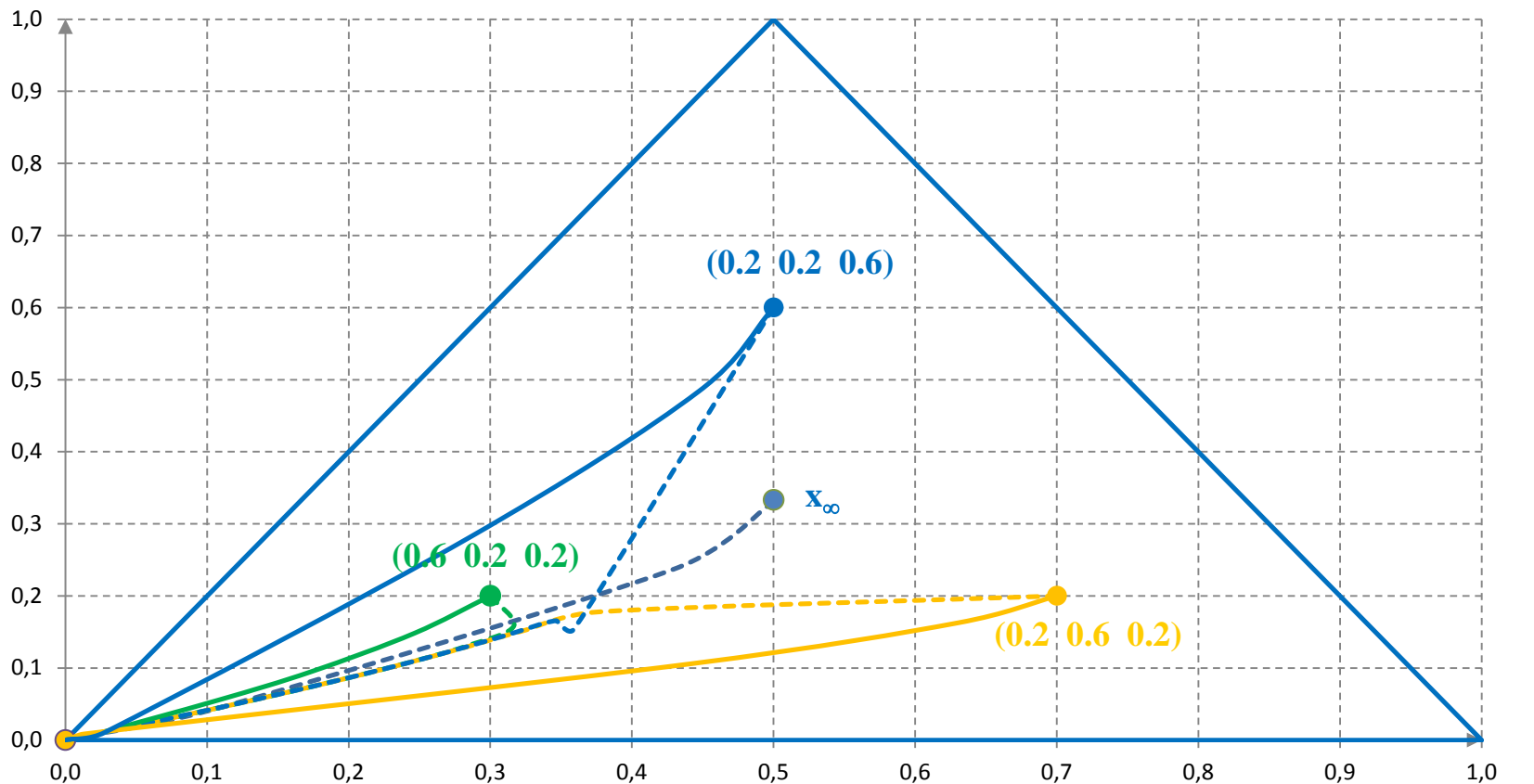
- Tracé dans le plan (A,B,C) $\rightarrow \begin{cases} u_1 = x_2 + x_3 / 2 \\ u_2 = x_3 \end{cases}$



3.2.3 Exemple

Comparaison

- Algorithme à pas long \rightarrow convergence beaucoup plus rapide (8 itérations au lieu de 45)
- Influence des réglages \rightarrow à adapter au cas par cas (valeurs de σ , δ , α)



3.2.4 Extensions

- ☐ Problème linéaire
- ☐ Problème quadratique
- ☐ Problème non linéaire
- ☐ Représentation

3.2.4 Problème linéaire

Problème linéaire

$$\min_{x \in \mathbb{R}^n} c^T x \text{ sous } \begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

- Lagrangien : $L(x, \lambda, s) = c^T x + \lambda^T (Ax - b) + s^T x$
- Conditions KKT : $\begin{cases} Ax - b = 0 \\ \nabla_x L(x, \lambda, s) = 0 \\ XS = 0 \\ x, s \geq 0 \end{cases} \rightarrow F(x, \lambda, s) = \begin{pmatrix} Ax - b \\ \nabla_x L(x, \lambda, s) \\ XSe - he \end{pmatrix} = 0 \rightarrow \text{barrière } h$
- Méthode de Newton : $\nabla F(x, \lambda, s) \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = -F(x, \lambda, s) \rightarrow \text{direction } \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix}$

$$\Rightarrow \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = - \begin{pmatrix} Ax - b \\ A^T \lambda + s + c \\ XSe - he \end{pmatrix} \begin{matrix} = 0 \\ = 0 \end{matrix} \text{ si admissible}$$
- On peut appliquer le même algorithme à des problèmes non linéaires.

3.2.4 Problème quadratique

Problème quadratique

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + c^T x \quad \text{sous} \quad \begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

- Lagrangien : $L(x, \lambda, s) = \frac{1}{2} x^T Q x + c^T x + \lambda^T (Ax - b) + s^T x$
- Conditions KKT : $\begin{cases} Ax - b = 0 \\ \nabla_x L(x, \lambda, s) = 0 \\ XS = 0 \\ x, s \geq 0 \end{cases} \rightarrow F(x, \lambda, s) = \begin{pmatrix} Ax - b \\ \nabla_x L(x, \lambda, s) \\ XSe - he \end{pmatrix} = 0 \rightarrow \text{barrière } h$
- Méthode de Newton : $\nabla F(x, \lambda, s) \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = -F(x, \lambda, s) \rightarrow \text{direction} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix}$

$$\Rightarrow \begin{pmatrix} A & 0 & 0 \\ Q & A^T & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = - \begin{pmatrix} Ax - b \\ Qx + A^T \lambda + s + c \\ XSe - he \end{pmatrix}$$

- Différence avec le cas linéaire : matrice $Q = \nabla_{xx}^2 L(x, \lambda, s)$

3.2.4 Problème non linéaire

Problème non linéaire

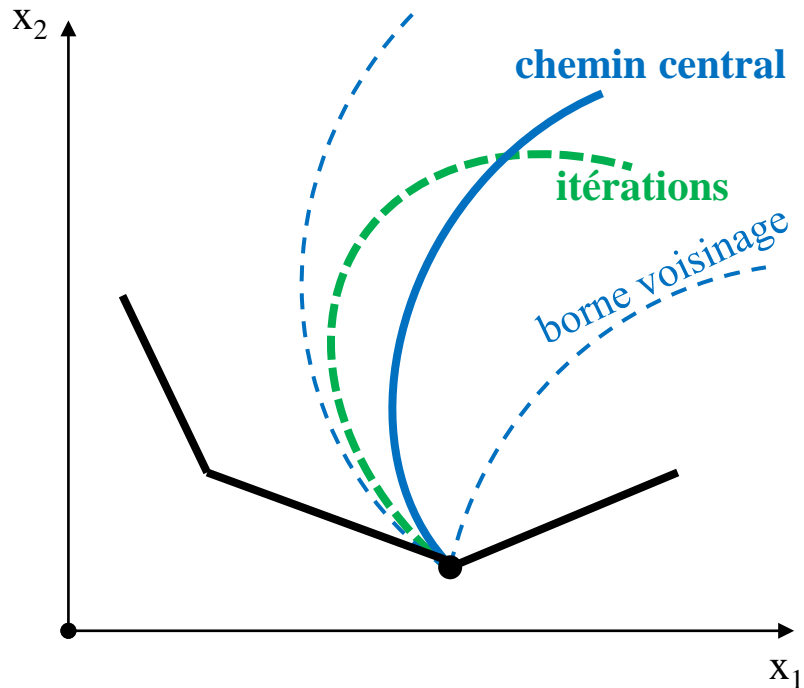
$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c(x) = 0 \\ x \geq 0 \end{cases} \rightarrow \text{contraintes actives}$$

- Lagrangien : $L(x, \lambda, s) = f(x) + \lambda^T c(x) + s^T x$
 - Conditions KKT : $\begin{cases} c(x) = 0 \\ \nabla_x L(x, \lambda, s) = 0 \\ XS = 0 \\ x, s \geq 0 \end{cases} \rightarrow F(x, \lambda, s) = \begin{pmatrix} c(x) \\ \nabla_x L(x, \lambda, s) \\ XSe - he \end{pmatrix} = 0 \rightarrow \text{barrière } h$
 - Méthode de Newton : $\nabla F(x, \lambda, s) \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = -F(x, \lambda, s) \rightarrow \text{direction } \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix}$
- $$\Rightarrow \begin{pmatrix} \nabla c(x)^T & 0 & 0 \\ \nabla_{xx}^2 L(x, \lambda, s) & \nabla c(x) & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \\ d_s \end{pmatrix} = - \begin{pmatrix} c(x) \\ \nabla_x L(x, \lambda, s) \\ XSe - he \end{pmatrix}$$
- Différence avec le cas linéaire : matrices $\begin{cases} \nabla_{xx}^2 L(x, \lambda, s) \\ \nabla c(x) \end{cases}$

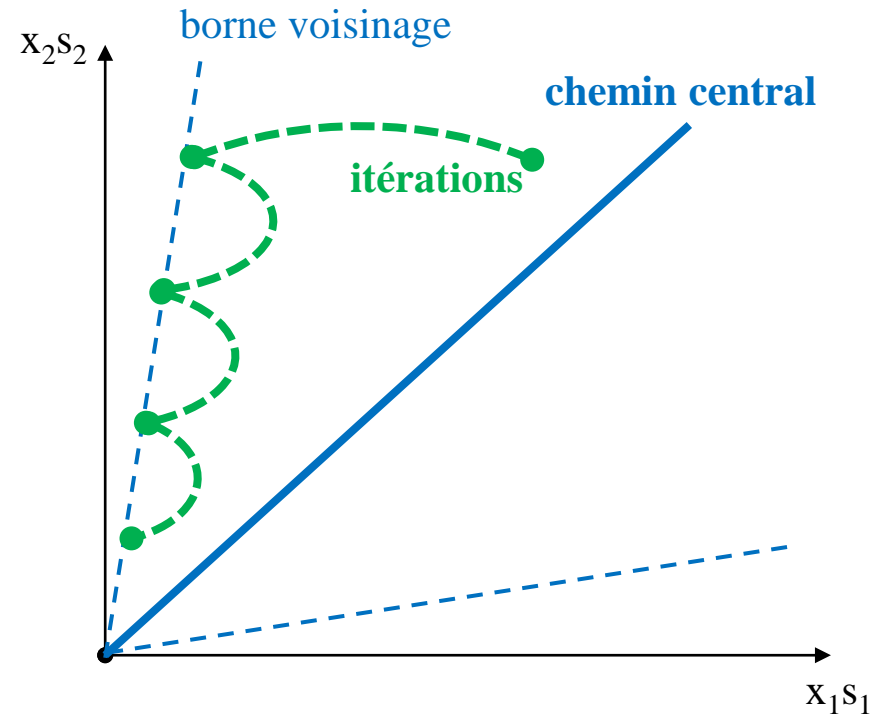
3.2.4 Représentation

Tracé des itérations

Espace primal : variables x



Espace primal-dual : produits xs



Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
- 3. Optimisation avec contraintes**
 - 3.1 Simplexe
 - 3.2 Point intérieur
 - 3.3 Gradient projeté**
 - 3.3.1 Principes
 - 3.3.2 Direction de déplacement
 - 3.3.3 Restauration
 - 3.3.4 Algorithme
 - 3.4 Lagrangien augmenté
 - 3.5 Programmation quadratique séquentielle
 - 3.6 Convergence

3.3.1 Gradient projeté

Problème avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \rightarrow m \text{ contraintes actives}$$

Etapes principales

A chaque itération

- Construction d'une direction de descente d_k à partir du point x_k
- Réglage du pas de déplacement s_k suivant d_k

Direction de descente

On construit la direction d_k dans l'hyperplan tangent aux contraintes (= espace nul) en x_k

- **Gradient projeté** → projection du gradient sur l'hyperplan tangent
- **Gradient réduit** → réduction du gradient sur une base de l'espace nul

Pas de déplacement

- Recherche linéaire suivant d_k → pas s_k
- Restauration de l'admissibilité → méthode de Newton
- Règles d'acceptation du pas → Armijo, Goldstein, Wolfe

3.3.2 Direction de déplacement

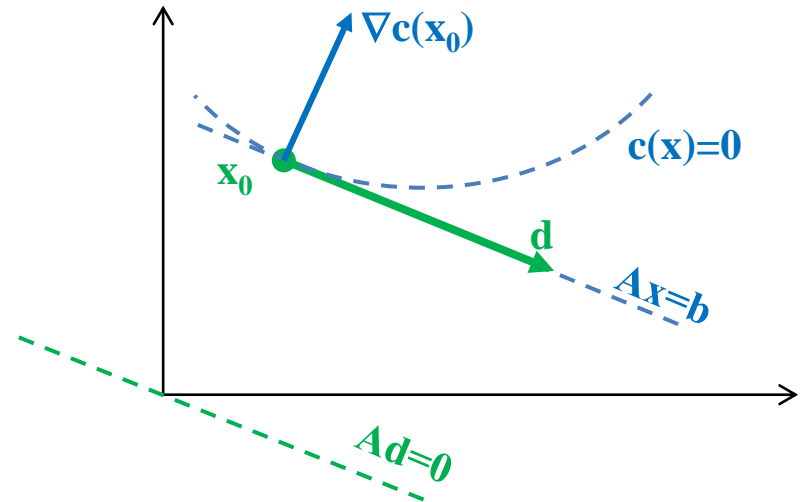
- ☐ Hyperplan tangent aux contraintes
- ☐ Gradient projeté
- ☐ Gradient réduit
- ☐ Exemple

3.3.2 Hyperplan tangent

Problème avec contraintes linéaires

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } Ax = b$$

- x_0 point admissible $\rightarrow Ax_0 = b$
- d déplacement admissible à partir de x_0 $\rightarrow A(x_0 + d) = b$ $\left. \vphantom{\begin{matrix} \rightarrow Ax_0 = b \\ \rightarrow A(x_0 + d) = b \end{matrix}} \right\} \rightarrow Ad = 0$
- $Ad = 0$ définit l'espace nul des contraintes
= hyperplan des contraintes
- Le déplacement $d \in \mathbb{R}^n$ est admissible si d est dans l'hyperplan des contraintes.



Problème avec contraintes non linéaires

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

- On définit l'**espace nul tangent** ou **hyperplan tangent** en x_0 avec $A = \nabla c(x_0)^T \rightarrow Ad = 0$
- On cherche un déplacement $d \in \mathbb{R}^n$ dans l'hyperplan tangent : $\nabla c(x_0)^T d = 0$
 Un déplacement complémentaire est ensuite nécessaire pour restaurer l'admissibilité.

3.3.2 Gradient projeté

Définition

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

Le gradient projeté est la **projection orthogonale** du gradient de f sur l'hyperplan tangent.

Expression du gradient projeté

- Hyperplan tangent aux contraintes en x_0 admissible
 $Ad = 0$ avec $A = \nabla c(x_0)^T$

- Matrice de projection sur l'hyperplan tangent

$$P = I - A^T (AA^T)^{-1} A$$

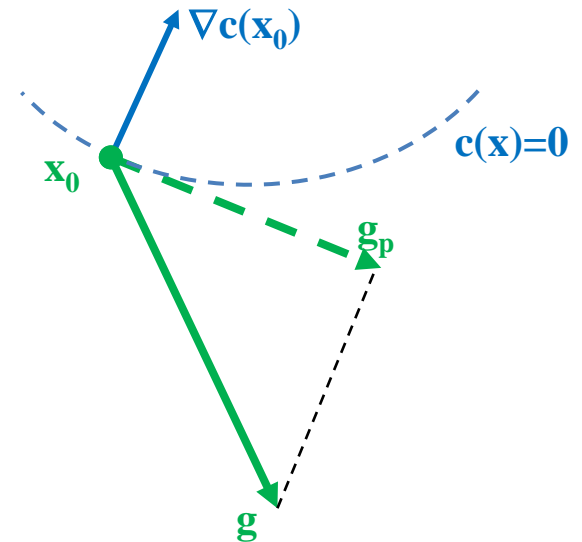
- Notations**

$$g(x_0) \text{ gradient de } f \text{ en } x_0 \rightarrow g(x_0) = \nabla f(x_0)$$

$$g_p(x_0) \text{ gradient projeté} \rightarrow g_p(x_0) = Pg(x_0)$$

$$\rightarrow g_p = \left(I - A^T (AA^T)^{-1} A \right) g \text{ avec } A = \nabla c(x_0)^T$$

- g_p vérifie : $\begin{cases} Ag_p = 0 \\ g_p^T (g - g_p) = 0 \end{cases} \rightarrow g_p \in \text{hyperplan tangent}$
 $\rightarrow g - g_p \perp \text{hyperplan tangent}$



$$\text{car } \begin{cases} P^T = P \\ P^2 = P \end{cases}$$

3.3.2 Gradient projeté

Direction de descente

- La direction du gradient projeté est la **direction de plus forte pente dans l'hyperplan tangent**
 = direction dans l'hyperplan qui maximise la dérivée directionnelle de f

Preuve

La direction d dans l'hyperplan maximisant la dérivée directionnelle de f est solution de

$$\min_{d \in \mathbb{R}^n} g^T d \quad \text{sous} \quad \begin{cases} Ad = 0 & \rightarrow d \in \text{hyperplan tangent} \\ \|d\| = 1 \Leftrightarrow d^T d = 1 & \rightarrow \text{norme} = 1 \end{cases}$$

Lagrangien : $L(d, \lambda, \mu) = g^T d + \lambda^T Ad + \mu(d^T d - 1)$ avec $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}$

$$\text{Conditions KKT : } \begin{cases} g + A^T \lambda + 2\mu d = 0 & \rightarrow d = -(g + A^T \lambda) / (2\mu) \\ Ad = 0 & \rightarrow Ag + AA^T \lambda = 0 \rightarrow \lambda = -(AA^T)^{-1} Ag \\ \|d\| = 1 & \rightarrow 2\mu = \pm \|g + A^T \lambda\| \end{cases}$$

d est bien un vecteur normé colinéaire à $(I - A^T (AA^T)^{-1} A)g$

- La méthode du gradient projeté équivaut à la méthode de plus forte pente appliquée dans l'espace nul des contraintes \rightarrow méthode d'ordre 1 peu efficace
 \rightarrow amélioration par méthode de quasi-Newton

3.3.2 Gradient réduit

Définition

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

Le gradient réduit est le gradient de la **fonction réduite** sur une base de l'espace nul tangent.

Expression du gradient réduit

- Base Z de l'espace nul tangent aux contraintes : $AZ = 0$ avec $A = \nabla c(x_0)^T$
- Décomposition du déplacement : $p = Yp_Y + Zp_Z$ avec $\begin{cases} AZ = 0 \\ AY \text{ inversible} \end{cases}$
- Déplacement admissible : $Ap = 0 \Rightarrow \begin{cases} p_Y = 0 \\ p = Zp_Z \end{cases}$
- Fonction réduite f_r : $\min_{p \in \mathbb{R}^n} f(x_0 + p) \text{ sous } A(x_0 + p) = b \Leftrightarrow \min_{p_Z \in \mathbb{R}^{n-m}} f_r(p_Z) = f(x_0 + Zp_Z)$
- **Notations**
 - $g(x_0)$ gradient de f en $x_0 \rightarrow g(x_0) = \nabla f(x_0) \rightarrow g \in \mathbb{R}^n$
 - $g_r(x_0)$ gradient réduit $\rightarrow g_r(x_0) = \nabla f_r(p_Z=0) \rightarrow g_r \in \mathbb{R}^{n-m}$ (m = nombre de contraintes)
 - $f_r(p_Z) = f(x_0 + Zp_Z) \Rightarrow \nabla f_r(p_Z) = Z^T \nabla f(x_0 + Zp_Z) \rightarrow \boxed{g_r = Z^T g}$ en $p_Z = 0$
 - $\rightarrow g_r$ est le gradient de la fonction réduite f_r (= fonction de $n-m$ variables p_Z)

3.3.2 Gradient réduit

Direction de descente

- Le déplacement à partir du point x_0 admissible est décomposé en

$$p = Yp_Y + Zp_Z \quad \text{avec} \quad \begin{cases} AZ = 0 \\ AY \text{ inversible} \end{cases} \quad Ap = 0 \Rightarrow \begin{cases} p_Y = 0 \\ p = Zp_Z \end{cases}$$

- Le gradient réduit g_r donne la direction **de plus forte pente suivant les variables p_Z** .
La direction de déplacement dans R^n est : **$d = Zg_r$** .
- On peut choisir les matrices Y et Z
 - à partir de matrices orthogonales \rightarrow factorisation QR de A
 - à partir d'une base de A $\rightarrow B \in R^{m \times m}$ (= m colonnes indépendantes de A)

Gradient réduit sur une base B de A

$$AE = \begin{pmatrix} m & n-m \\ B & N \end{pmatrix} \Rightarrow g = \begin{pmatrix} g_B \\ g_N \end{pmatrix} \quad Y = \begin{pmatrix} B^{-1} \\ 0 \end{pmatrix}_{n-m}^m \quad Z = \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix}_{n-m}^{n-m}$$

(E = matrice de permutation de colonnes de A)

- Le gradient réduit par rapport à la base B est : $g_r = Z^T g = g_N - (B^{-1}N)^T g_B$

3.3.2 Direction de déplacement

Problème avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow \text{ m contraintes actives}$$

- On construit la direction de déplacement $d \in \mathbb{R}^n$ dans l'hyperplan tangent aux contraintes en x_0
 $Ad = 0$ avec $A = \nabla c(x_0)^T$
 \rightarrow 2 méthodes de construction de la direction d

- Méthode du gradient projeté**

La direction d est celle du gradient projeté : $d = g_p$

$$d = Pg \text{ avec } P = I - A^T(AA^T)^{-1}A \quad (P = \text{matrice de projection sur l'hyperplan tangent})$$

- Méthode du gradient réduit**

La direction d est obtenue à partir du gradient de la fonction réduite : $d = Zg_r$ avec $g_r = Z^T g$

$$d = ZZ^T g \text{ avec } AZ = 0 \quad (Z = \text{base de l'hyperplan tangent})$$

- On cherche un pas de déplacement suivant $-d$ pour minimiser f .
 Un déplacement complémentaire est nécessaire pour restaurer l'admissibilité.

3.3.2 Exemple

Exemple

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

$$\nabla f(x) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \nabla c(x) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 1) \end{pmatrix}$$

- Changement de variables (coordonnées polaires)

$$\begin{cases} x_1 = r \cos \theta \\ x_2 = r \sin \theta + 1 \end{cases} \rightarrow \begin{cases} f(r, \theta) = r(\cos \theta + \sin \theta) + 1 \\ c(r, \theta) = r^2 - 1 \end{cases}$$

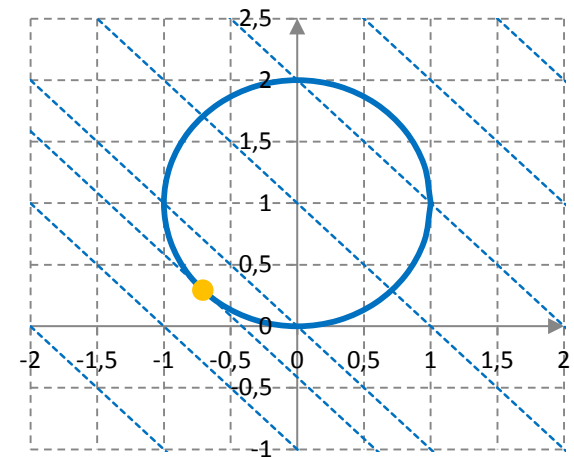
- Elimination variable r

$$r = 1 \rightarrow f(\theta) = \cos \theta + \sin \theta + 1$$

- Minimum**

$$\left\{ \begin{array}{l} f'(\theta) = -\sin \theta + \cos \theta = 0 \rightarrow \tan \theta = 1 \rightarrow \theta = \frac{\pi}{4} \text{ ou } \frac{\pi}{4} + \pi \\ f''(\theta) = -\cos \theta - \sin \theta \geq 0 \rightarrow -\cos \theta(1 + \tan \theta) \geq 0 \rightarrow -\cos \theta \geq 0 \end{array} \right\} \rightarrow \theta^* = \frac{\pi}{4} + \pi$$

$$\rightarrow \begin{cases} x_1^* = -1/\sqrt{2} \approx -0.70711 \\ x_2^* = 1 - 1/\sqrt{2} \approx 0.29289 \end{cases}$$



3.3.2 Exemple

Gradient projeté

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

Point admissible $x_0 (r_0=1, \theta_0) \rightarrow \nabla f(x_0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\nabla c(x_0) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 1) \end{pmatrix} = 2r_0 \begin{pmatrix} \cos \theta_0 \\ \sin \theta_0 \end{pmatrix}$

- Gradient projeté au point x_0

$$g_p = Pg \text{ avec } \begin{cases} A = \nabla c(x_0)^T, g = \nabla f(x_0) \\ P = I - A^T(AA^T)^{-1}A \end{cases}$$

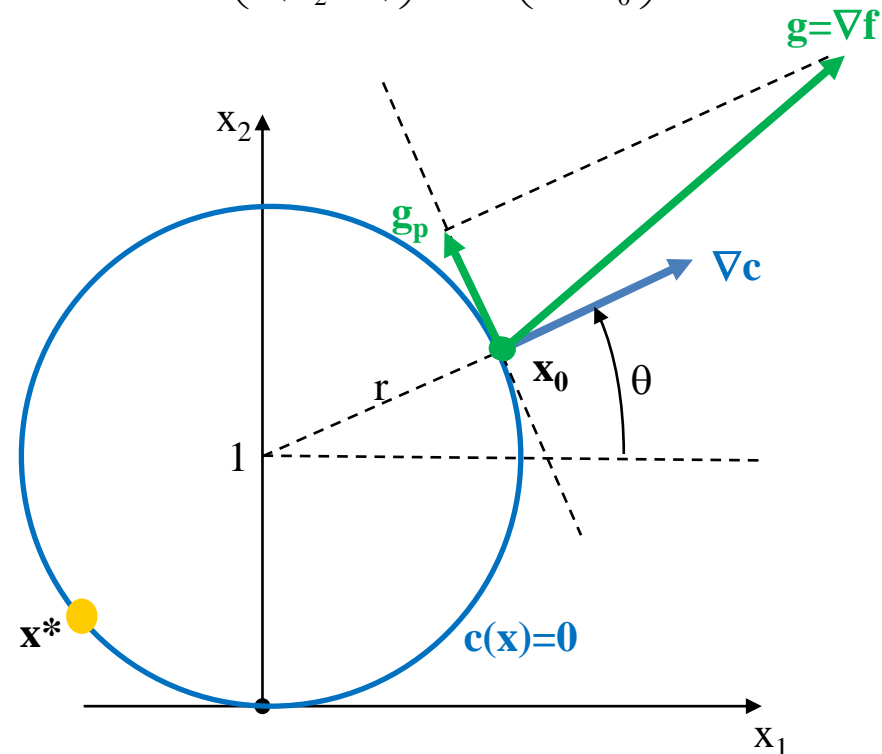
$$A = 2r_0(\cos \theta_0 \quad \sin \theta_0)$$

$$P = \begin{pmatrix} \sin^2 \theta_0 & \sin \theta_0 \cos \theta_0 \\ \sin \theta_0 \cos \theta_0 & \cos^2 \theta_0 \end{pmatrix}$$

$$\rightarrow g_p = (\cos \theta_0 - \sin \theta_0) \begin{pmatrix} -\sin \theta_0 \\ \cos \theta_0 \end{pmatrix}$$

- Direction de descente au point x_0

$$d = \frac{g_p}{\|g_p\|} = \begin{pmatrix} -\sin \theta_0 \\ \cos \theta_0 \end{pmatrix} \rightarrow \text{tangente au cercle en } x_0$$



3.3.2 Exemple

Gradient réduit

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

Point admissible $x_0 (r_0=1, \theta_0) \rightarrow \nabla f(x_0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\nabla c(x_0) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 1) \end{pmatrix} = 2r_0 \begin{pmatrix} \cos \theta_0 \\ \sin \theta_0 \end{pmatrix}$

- Gradient réduit au point x_0

$$g_r = Z^T g \text{ avec } \begin{cases} A = \nabla c(x_0)^T, g = \nabla f(x_0) \\ AE = (B \quad N), Z = \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix} \end{cases}$$

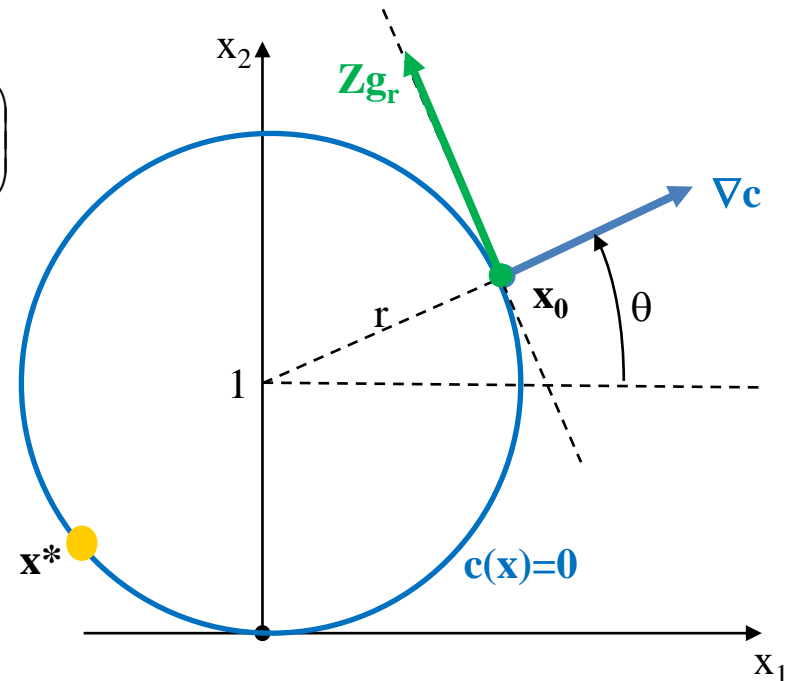
$$A = 2r_0 (\cos \theta_0 \quad \sin \theta_0) \quad Z = \begin{pmatrix} -\tan \theta_0 \\ 1 \end{pmatrix}$$

$$\rightarrow g_r = 1 - \tan \theta_0$$

$$\rightarrow Zg_r = \frac{\cos \theta_0 - \sin \theta_0}{\cos \theta_0} \begin{pmatrix} -\sin \theta_0 \\ \cos \theta_0 \end{pmatrix}$$

- Direction de descente au point x_0

$$d = \frac{Zg_r}{\|Zg_r\|} = \begin{pmatrix} -\sin \theta_0 \\ \cos \theta_0 \end{pmatrix} \rightarrow \text{tangente au cercle en } x_0$$



3.3.3 Restauration

- ☐ Point initial
- ☐ Itérations admissibles
- ☐ Méthode de restauration

3.3.3 Restauration

Itérations admissibles

La méthode du gradient projeté ou réduit construit une suite de solutions admissibles
 → point initial admissible
 → restauration de la faisabilité à chaque itération

Point initial

- On peut construire un point initial admissible du problème $\min_{x \in \mathbb{R}^n} f(x)$ sous $\begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$
 en résolvant le **problème préliminaire sans contrainte** $\min_{x \in \mathbb{R}^n} \|c_E(x)\|_2 + \|\max(0, c_I(x))\|_2$
- La solution x_0 de ce problème préliminaire est admissible si le coût est nul

$$\|c_E(x_0)\|_2 + \|\max(0, c_I(x_0))\|_2 = 0 \Rightarrow \begin{cases} c_E(x_0) = 0 \\ \max(0, c_I(x_0)) = 0 \end{cases} \Rightarrow \begin{cases} c_E(x_0) = 0 \\ c_I(x_0) \leq 0 \end{cases}$$

Restauration de la faisabilité

- La direction de descente d est dans l'hyperplan tangent aux contraintes au point courant.
- Si les contraintes ne sont pas linéaires, un pas s suivant d donne un point non admissible
 → **Il faut restaurer la faisabilité avant d'évaluer si le pas s est acceptable.**

3.3.3 Restauration

Déplacement admissible

$\min_{x \in \mathbb{R}^n} f(x)$ sous $c(x) = 0 \rightarrow m$ contraintes actives

On construit le déplacement p à partir du point initial x_0 en 2 étapes : $p = p_1 + p_2$

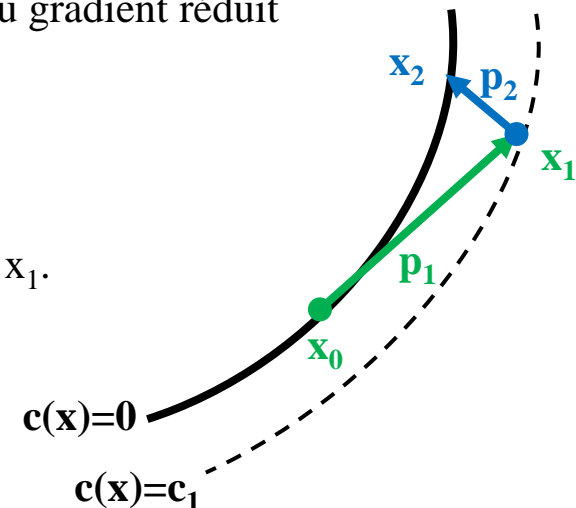
- Le déplacement p_1 est suivant la direction de descente d dans l'hyperplan tangent : $p_1 = -sd$
 $d \in \mathbb{R}^n$ = direction construite à partir du gradient projeté ou du gradient réduit
 $s > 0$ = pas de déplacement suivant $-d$ (pour minimisation)

On obtient un point $x_1 = x_0 + p_1$ dans l'hyperplan tangent

\rightarrow non admissible si contraintes non linéaires

- Le déplacement p_2 restaure un point admissible à partir du point x_1 .
 \rightarrow linéarisation des contraintes en x_1
 \rightarrow résolution d'un système sous-déterminé

On obtient un point $x_2 = x_1 + p_2$ admissible.



Recherche linéaire

- On évalue le point x_2 correspondant au pas s de recherche linéaire suivant d .
- Le pas s est modifié par dichotomie jusqu'à trouver un point $x_2(s)$ acceptable
 \rightarrow règles d'Armijo, Goldstein, Wolfe,...

3.3.3 Restauration

Méthode de restauration

Le déplacement p_2 doit vérifier : $A_1 p = b_1$ avec $\begin{cases} A_1 = \nabla c(x_1)^T \approx \nabla c(x_0)^T = A_0 \\ b_1 = -c(x_1) = -c_1 \end{cases}$

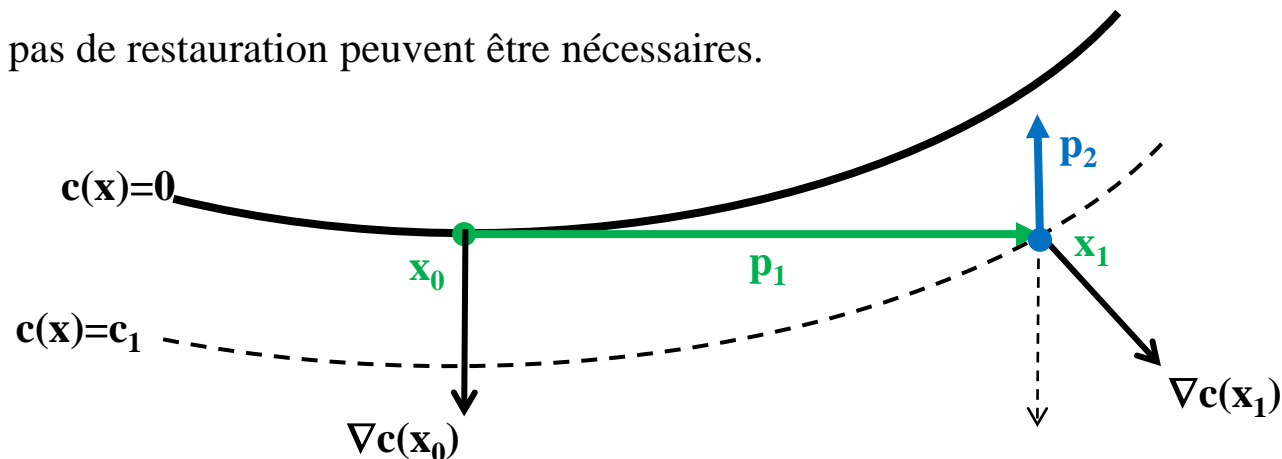
- **Solution de norme minimale** → projection sur l'hyperplan tangent aux contraintes actives (cf §1.2.4)

$$\min_{p \in \mathbb{R}^n} \|p\| \text{ sous } A_1 p = b_1 \quad \rightarrow \quad p_2 = A_1^T (A_1 A_1^T)^{-1} b_1$$

- **Solution de base** → pour ne pas dégrader la minimisation apportée par p_1 (cf §1.2.3)

$$A_1 (Y p_Y + Z p_Z) = b_1 \Rightarrow p_Y = (A_1 Y)^{-1} b_1 \rightarrow p_2 = Y (A_1 Y)^{-1} b_1$$

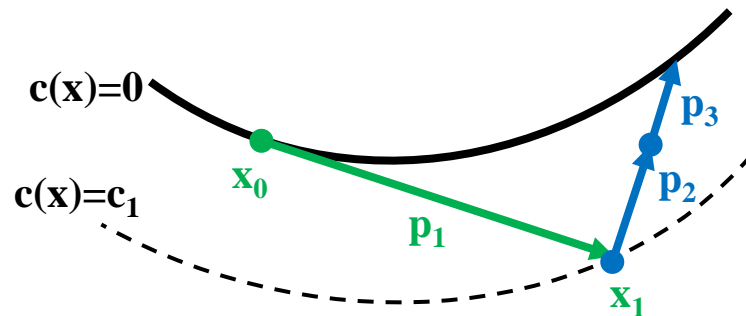
- Plusieurs pas de restauration peuvent être nécessaires.



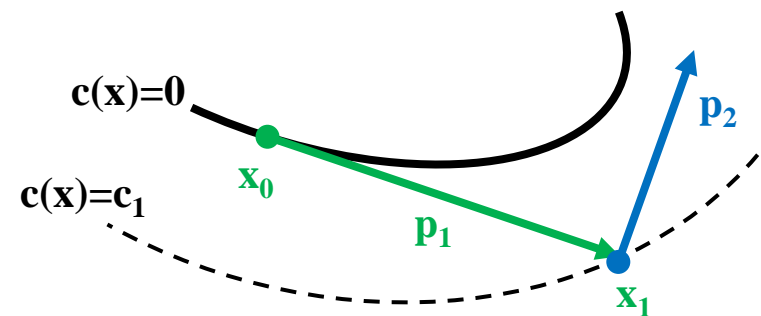
3.3.3 Restauration

Illustrations

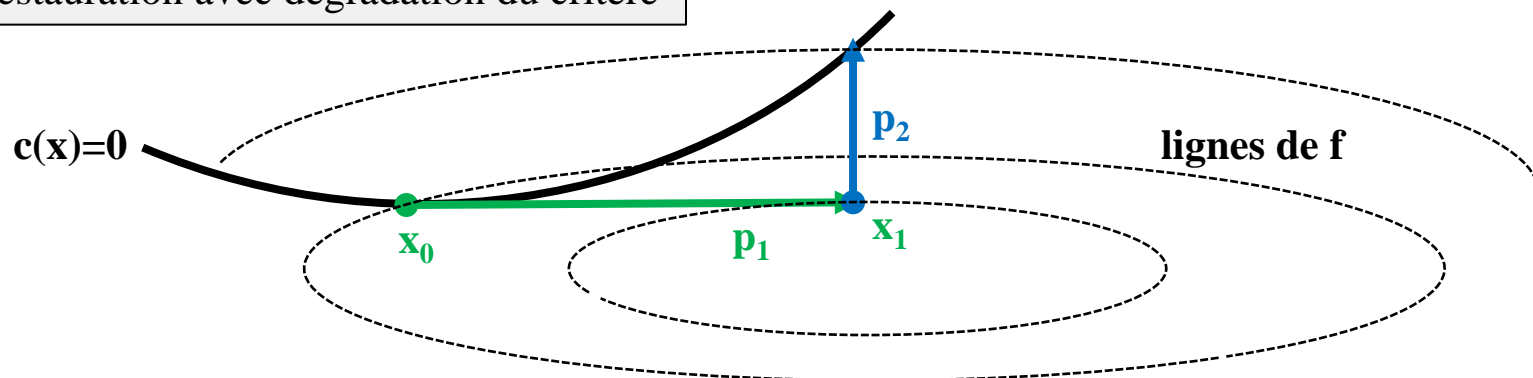
Restauration en plusieurs itérations : p_2, p_3



Restauration infructueuse (non linéarité)



Restauration avec dégradation du critère



3.3.4 Algorithme

- ❑ Algorithme de gradient projeté / réduit
- ❑ Exemple

3.3.4 Algorithme

Algorithme de gradient projeté/réduit

Les principales étapes d'une itération de gradient projeté/réduit sont

- construire la direction de descente au point courant
- effectuer une recherche linéaire avec restauration

Direction de descente

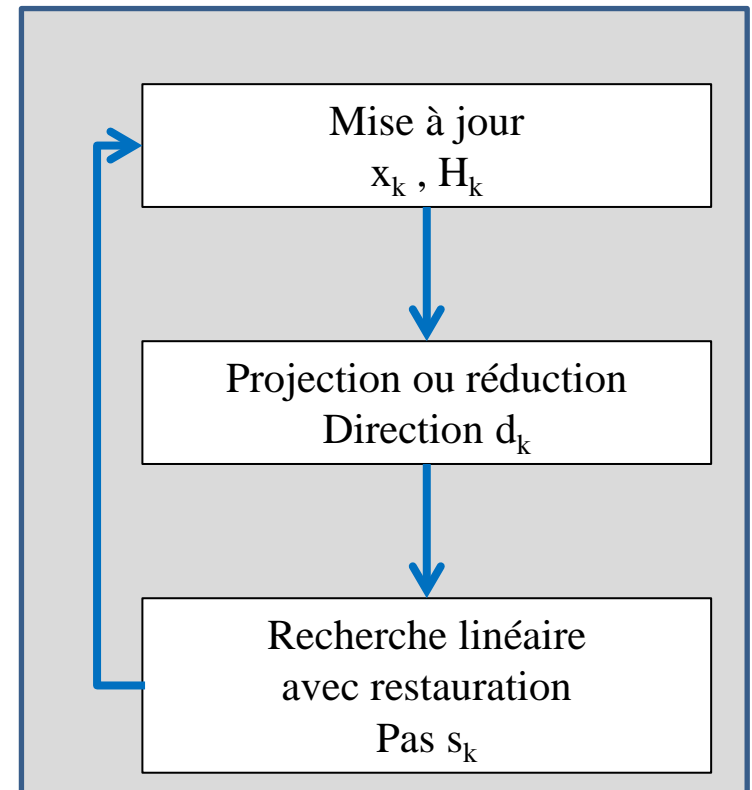
- Sélection des contraintes actives
- Projection ou réduction dans l'hyperplan tangent
- Mise à jour du hessien (quasi-Newton)

Recherche linéaire

- Méthode de dichotomie sur le pas de déplacement
- Restauration avant évaluation du pas
- Règles d'acceptation (Armijo,...)

Principales difficultés

- Amélioration critère → grands pas
- Restauration contraintes → petits pas
→ difficultés sur problèmes très non-linéaires
→ réglages à adapter au cas par cas



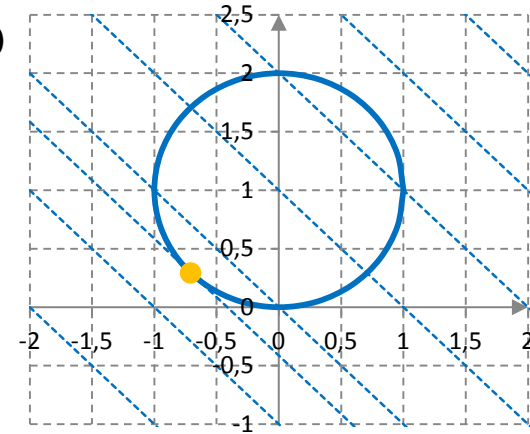
3.3.4 Exemple

Exemple

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$
- Solution :
$$\begin{cases} x_1^* = -1/\sqrt{2} \approx -0.70711 \\ x_2^* = 1 - 1/\sqrt{2} \approx 0.29289 \end{cases}$$

Itérations

- Point courant : $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} r \cos \theta \\ r \sin \theta + 1 \end{pmatrix}$
- Descente : $x' = x - s_1 d_1$ avec $d_1 = \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix} \leftarrow g_p = (\cos \theta - \sin \theta) \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}$
 → **pas s_1 suivant le gradient projeté**
- Restauration : $x'' = x' - s_2 d_2$ avec $d_2 = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \leftarrow \nabla c(x) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 1) \end{pmatrix}$
 → **pas s_2 suivant le gradient des contraintes**
- Réglage des pas : s_2 est calculé pour restaurer $c(x'') = 0$
 s_1 est choisi pour vérifier une décroissance suffisante $f(x'') < f(x)$



3.3.4 Exemple

Exemple

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$
- Point initial : $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} \rightarrow$ Restauration initiale : $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Itération	x_1	x_2	$f(x)$	$c(x)$	Descente s_1	x_1'	x_2'	$c(x')$	Restauration s_2
1	0,10000	1,00000	1,10000	-0,99000	0,00000	0,10000	1,00000	-0,99000	4,50000
2	1,00000	1,00000	2,00000	0,00000	1,00000	1,00000	0,00000	1,00000	-0,50000
3	0,00000	0,00000	0,00000	0,00000	0,50000	-0,50000	0,00000	0,25000	-0,06699
4	-0,50000	0,13397	-0,36603	0,00000	0,18301	-0,65849	0,22548	0,03349	-0,00844
5	-0,65005	0,24011	-0,40994	0,00000	5,492E-02	-0,69178	0,27581	3,016E-03	-7,547E-04
6	-0,69080	0,27696	-0,41385	0,00000	1,612E-02	-0,70246	0,28809	2,599E-04	-6,497E-05
7	-0,70237	0,28819	-0,41418	0,00000	4,722E-03	-0,70573	0,29150	2,230E-05	-5,576E-06
8	-0,70572	0,29151	-0,41421	0,00000	1,383E-03	-0,70670	0,29249	1,913E-06	-4,783E-07
9	-0,70670	0,29249	-0,41421	0,00000	4,051E-04	-0,70699	0,29277	1,641E-07	-4,103E-08
10	-0,70699	0,29277	-0,41421	0,00000	1,187E-04	-0,70707	0,29286	1,408E-08	-3,520E-09
11	-0,70707	0,29286	-0,41421	0,00000	3,475E-05	-0,70710	0,29288	1,208E-09	-3,020E-10
12	-0,70710	0,29288	-0,41421	0,00000					

3.3.4 Exemple

Exemple

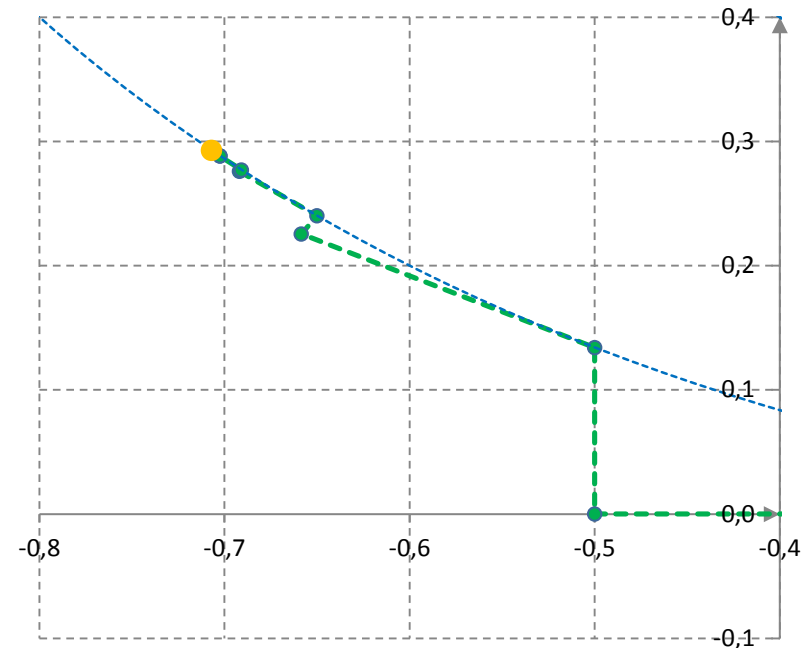
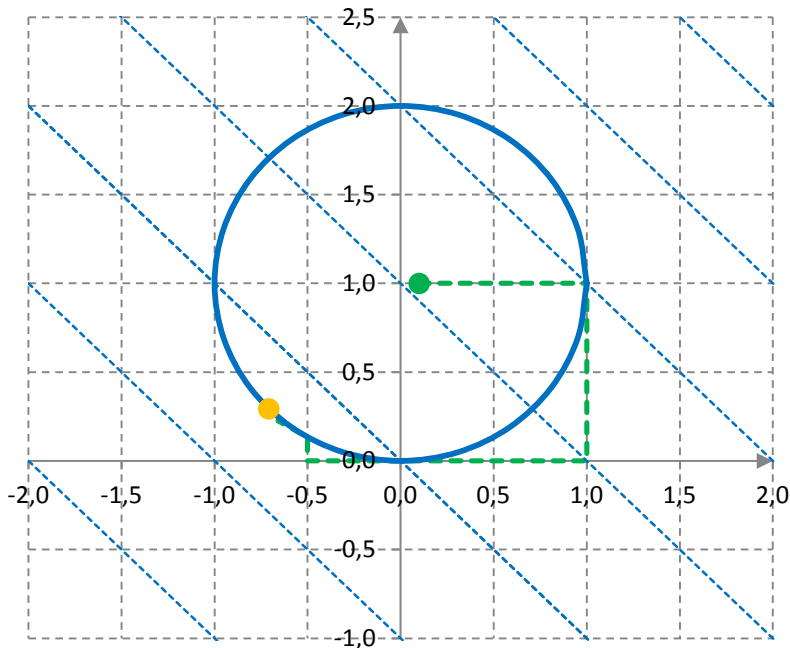
Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

Point initial

Point initial admissible

Solution

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 1 \end{pmatrix} \xrightarrow{\text{restauration initiale}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \xrightarrow{\text{itérations}} \begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix} = \begin{pmatrix} -1/\sqrt{2} \approx -0.70711 \\ 1 - 1/\sqrt{2} \approx 0.29289 \end{pmatrix}$$



Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
- 3. Optimisation avec contraintes**
 - 3.1 Simplexe
 - 3.2 Point intérieur
 - 3.3 Gradient projeté
 - 3.4 Lagrangien augmenté**
 - 3.4.1 Principes
 - 3.4.2 Pénalisation
 - 3.4.3 Algorithme
 - 3.5 Programmation quadratique séquentielle
 - 3.6 Convergence

3.4.1 Lagrangien augmenté

Problème avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \rightarrow \text{contraintes actives}$$

La difficulté de résolution vient des 2 objectifs antagonistes :

- Minimiser le critère $f(x)$
- Satisfaire les contraintes $c(x)=0$

Méthodes de pénalisation

Les contraintes sont ajoutées à la fonction coût avec une pondération :

- Critère augmenté \rightarrow pondération = pénalisation des contraintes
- Lagrangien \rightarrow pondération = multiplicateurs de Lagrange
- Lagrangien augmenté \rightarrow pondération = pénalisation + multiplicateurs

\rightarrow On se ramène à un **problème sans contraintes** plus simple

Les difficultés viennent du réglage de la pondération :

- Le problème pénalisé sans contraintes doit être équivalent au problème avec contraintes.
- Le problème pénalisé est mal conditionné lorsque la pénalisation est grande.

3.4.2 Pénalisation

- ☐ Critère augmenté
- ☐ Pénalisation quadratique
- ☐ Pénalisation exacte
- ☐ Mise en œuvre
- ☐ Lagrangien augmenté

3.4.2 Critère augmenté

Problème avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \rightarrow \text{contraintes actives}$$

On note x^* la solution du problème avec contraintes.

Critère augmenté

On ajoute au critère un terme positif fonction de la violation des contraintes avec un coefficient de pénalisation $\rho > 0 \rightarrow$ 2 méthodes usuelles de pénalisation

- **Pénalisation en norme 2** (pénalisation quadratique)

$$f_\rho(x) = f(x) + \frac{1}{2} \rho \left(\|c_E(x)\|_2^2 + \|\max(0, c_I(x))\|_2^2 \right) \Leftrightarrow f_\rho(x) = f(x) + \frac{1}{2} \rho \|c(x)\|_2^2$$

- **Pénalisation en norme 1**

$$f_\rho(x) = f(x) + \rho \left(\|c_E(x)\|_1 + \|\max(0, c_I(x))\|_1 \right) \Leftrightarrow f_\rho(x) = f(x) + \rho \|c(x)\|_1$$

Problème sans contraintes

$$\min_{x \in \mathbb{R}^n} f_\rho(x) \rightarrow \text{solution } x_\rho$$

3.4.2 Pénalisation quadratique

Problème pénalisé l_2

$$\min_{x \in \mathbb{R}^n} f_\rho(x) \quad \text{avec} \quad f_\rho(x) = f(x) + \frac{1}{2} \rho \left(\|c_E(x)\|_2^2 + \|\max(0, c_I(x))\|_2^2 \right)$$

$$\Leftrightarrow f_\rho(x) = f(x) + \frac{1}{2} \rho \|c(x)\|_2^2 \quad \rightarrow \text{contraintes actives}$$

Le critère l_2 est différentiable deux fois pour un problème avec contraintes égalité
 On peut appliquer les algorithmes d'optimisation sans contraintes à base de gradient.

Méthode de résolution

- On résout une suite de problèmes pénalisés avec des valeurs croissantes de la pénalisation ρ .
- Chaque problème $k+1$ est initialisé avec la solution précédente x_k .
- Problème k avec pénalisation ρ_k : $\min_{x \in \mathbb{R}^n} f_{\rho_k}(x) \rightarrow \text{solution } x_k$
- Il faut vérifier que la suite des solutions x_k converge vers la solution x^* du problème initial

$$\lim_{k \rightarrow \infty} x_k = x^* \quad \text{si} \quad \lim_{k \rightarrow \infty} \rho_k = +\infty$$

\rightarrow 2 résultats de convergence selon que x_k est une solution exacte ou approchée

3.4.2 Pénalisation quadratique

Problème pénalisé l_2

- Problème avec contraintes : $\min_{x \in \mathbb{R}^n} f(x)$ sous $c(x) = 0 \rightarrow$ solution x^*
- Problème k avec pénalisation ρ_k : $\min_{x \in \mathbb{R}^n} f_{\rho_k}(x)$ \rightarrow solution x_k
 $\lim_{k \rightarrow \infty} \rho_k = +\infty$ $\lim_{k \rightarrow \infty} x_k = x_\infty$

Convergence

- Si x_k est le **minimum global exact**, alors $\lim_{k \rightarrow \infty} x_k = x^*$
- Si x_k est un **minimum local approché** : $\|\nabla f_{\rho_k}(x_k)\| \leq \varepsilon_k$ avec $\lim_{k \rightarrow \infty} \varepsilon_k = 0$
 \rightarrow précision de résolution ε_k décroissante

alors la limite x_∞ est : - soit un point non admissible qui minimise $\|c(x)\|_2^2$
 - soit un point x^* vérifiant les conditions KKT du problème initial

On a dans ce 2^{ème} cas : $\begin{cases} \lim_{k \rightarrow \infty} x_k = x^* \\ \lim_{k \rightarrow \infty} \rho_k c(x_k) = \lambda^* \end{cases} \rightarrow$ minimum local
 \rightarrow multiplicateurs des contraintes actives

La solution exacte x^* n'est obtenue qu'à la limite lorsque la pénalisation ρ tend vers l'infini.

3.4.2 Pénalisation quadratique

Eléments de la démonstration

$$f_{\rho}(x) = f(x) + \frac{1}{2}\rho\|c(x)\|^2 \quad \Rightarrow \quad \nabla f_{\rho}(x) = \nabla f(x) + \rho \nabla c(x)c(x)$$

- Critère d'arrêt sur x_k : $\|\nabla f_{\rho_k}(x_k)\| \leq \varepsilon_k$

$$\|\nabla f_{\rho_k}(x_k)\| = \|\nabla f(x_k) + \rho_k \nabla c(x_k)c(x_k)\| \geq \|\rho_k \nabla c(x_k)c(x_k)\| - \|\nabla f(x_k)\| \quad \text{car } \|a+b\| \geq \|a\| - \|b\|$$

$$\Rightarrow \|\rho_k \nabla c(x_k)c(x_k)\| \leq \varepsilon_k + \|\nabla f(x_k)\| \quad \Rightarrow \quad \|\nabla c(x_k)c(x_k)\| \leq \frac{\varepsilon_k + \|\nabla f(x_k)\|}{\rho_k} \xrightarrow{k \rightarrow \infty} 0$$

$$\Rightarrow \begin{cases} \nabla c(x^*) = 0 \rightarrow \min \|c(x)\| \\ \text{ou} \\ c(x^*) = 0 \rightarrow \text{admissible} \end{cases} \quad \text{si les gradients sont linéairement indépendants}$$

- Multiplicateurs de Lagrange

$$\|\nabla f_{\rho_k}(x_k)\| = \|\nabla f(x_k) + \rho_k \nabla c(x_k)c(x_k)\| \leq \varepsilon_k \xrightarrow{k \rightarrow \infty} 0$$

$$\rightarrow \nabla f(x^*) + \nabla c(x^*)\lambda^* = 0 \quad \text{avec } \lambda^* = \lim_{k \rightarrow \infty} \rho_k c(x_k)$$

3.4.2 Pénalisation exacte

Problème pénalisé l_1

$$\min_{x \in \mathbb{R}^n} f_\rho(x) \quad \text{avec} \quad f_\rho(x) = f(x) + \rho \left(\|c_E(x)\|_1 + \|\max(0, c_I(x))\|_1 \right)$$

$$\Leftrightarrow f_\rho(x) = f(x) + \rho \|c(x)\|_1 \quad \rightarrow \text{contraintes actives}$$

Le critère l_1 n'est pas différentiable.

Méthode de résolution

- On résout une suite de problèmes pénalisés avec des valeurs croissantes de la pénalisation ρ .
- Chaque problème $k+1$ est initialisé avec la solution précédente x_k .
- Problème k avec pénalisation ρ_k : $\min_{x \in \mathbb{R}^n} f_{\rho_k}(x) \rightarrow \text{solution } x_k$

Convergence

- Si $\rho > \rho^* = \|\lambda^*\|_\infty = \max |\lambda_i|$ alors **x^* est un minimum local de f_ρ** avec la pénalisation l_1 .
- La pénalisation l_1 est **exacte** si ρ est supérieur au plus grand multiplicateur.
 → ne nécessite pas d'augmenter indéfiniment ρ pour obtenir la solution exacte x^*

3.4.2 Mise en oeuvre

Méthodes avec critère augmenté

- Type de pénalisation
 - Pénalisation l_2 → différentiable, mais nécessite une pénalisation forte pour approcher x^*
 - Pénalisation l_1 → exacte, mais non différentiable
- Réglage de la pénalisation
 - Trop faible → risque de divergence (pas de minimum du problème pénalisé)
 - Trop forte → mauvais conditionnement, difficultés numériques
- Utilisation du critère augmenté
 - Difficultés pratiques si l'on veut obtenir une bonne précision sur la solution x^*
 - Le critère augmenté peut servir de fonction mérite dans le cadre d'autres algorithmes pour évaluer la progression vers l'optimum.

Méthodes avec lagrangien augmenté

On cherche à se ramener à une suite de problèmes sans contraintes

- en conservant un critère différentiable
 - en évitant le mauvais conditionnement dû à une pénalisation trop forte
- utilisation des multiplicateurs de Lagrange pour réduire la pénalisation

3.4.2 Lagrangien augmenté

Problème pénalisé l_2

- La méthode de pénalisation consiste à minimiser le critère augmenté.

$$\min_{x \in \mathbb{R}^n} f_\rho(x) = f(x) + \frac{1}{2} \rho \|c(x)\|^2$$

- La convergence est obtenue pour des valeurs croissantes de pénalisation.

$$\lim_{k \rightarrow \infty} \rho_k = +\infty \rightarrow \begin{cases} \lim_{k \rightarrow \infty} x_k = x^* & \rightarrow \text{minimum local} \\ \lim_{k \rightarrow \infty} \rho_k c(x_k) = \lambda^* & \rightarrow \text{multiplicateurs des contraintes actives} \end{cases}$$

- La solution x_k ne respecte qu'approximativement les contraintes : $c(x_k) \approx \frac{\lambda^*}{\rho_k}$
- Pour respecter précisément les contraintes, il faut augmenter fortement la pénalisation
 → cause de mauvais conditionnement et de difficultés numériques
- On peut appliquer la méthode de pénalisation **au problème équivalent**

$$\min_{x \in \mathbb{R}^n} L(x, \lambda^*) \text{ sous } c(x) = 0 \Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0$$

→ si l'on connaît λ^*

$$\rightarrow \min_{x \in \mathbb{R}^n} L_\rho(x, \lambda^*) = L(x, \lambda^*) + \frac{1}{2} \rho \|c(x)\|_2^2 = f(x) + \lambda^{*T} c(x) + \frac{1}{2} \rho \|c(x)\|^2$$

3.4.2 Lagrangien augmenté

Lagrangien augmenté

- La méthode de lagrangien augmenté consiste à résoudre une suite de problèmes :

$$\min_{x \in \mathbb{R}^n} L_{\rho_k}(x, \lambda_k) = L(x, \lambda_k) + \frac{1}{2} \rho_k \|c(x)\|_2^2 = f(x) + \lambda_k^T c(x) + \frac{1}{2} \rho_k \|c(x)\|_2^2$$

avec ρ_k = valeur de pénalisation du problème k

λ_k = estimation des multiplicateurs pour le problème k

- Si $\lim_{k \rightarrow \infty} \lambda_k = \lambda^*$ et $\lim_{k \rightarrow \infty} \rho_k = +\infty$ les problèmes deviennent équivalents.

$$\left. \begin{aligned} \min_{x \in \mathbb{R}^n} L(x, \lambda^*) \text{ sous } c(x) = 0 &\Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \\ \Leftrightarrow \min_{x \in \mathbb{R}^n} L_{\rho}(x, \lambda^*) &\Leftrightarrow \min_{x \in \mathbb{R}^n} f_{\rho}(x) \end{aligned} \right\} \rightarrow \lim_{k \rightarrow \infty} x_k = x^*$$

La solution x_k du problème $\min_{x \in \mathbb{R}^n} L_{\rho_k}(x, \lambda_k)$ converge vers la solution x^* du problème initial.

- La solution x_k vérifie également : $\nabla_x L_{\rho_k}(x_k, \lambda_k) = \nabla f(x_k) + \nabla c(x_k)(\lambda_k + \rho_k c(x_k)) = 0$

à comparer à x^* qui vérifie : $\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) + \nabla c(x^*) \lambda^* = 0$

$$\Rightarrow \lim_{k \rightarrow \infty} \lambda_k + \rho_k c(x_k) = \lambda^*$$

(même démonstration que pour le critère augmenté avec pénalisation quadratique)

3.4.2 Lagrangien augmenté

Lagrangien augmenté

- On peut estimer les multiplicateurs à l'itération k

$$\lim_{k \rightarrow \infty} \lambda_k + \rho_k c(x_k) = \lambda^* \rightarrow \lambda^* \approx \lambda_k + \rho_k c(x_k) \quad \text{pour } \rho_k \text{ assez grand}$$

$$\rightarrow \lambda_{k+1} = \lambda_k + \rho_k c(x_k) \quad \text{pour l'itération } k+1$$

- La valeur des contraintes à l'itération k est : $c(x_k) \approx \frac{\lambda^* - \lambda_k}{\rho_k} \xrightarrow{\lambda_k \rightarrow \lambda^*} 0$

On peut parvenir à respecter les contraintes sans augmenter indéfiniment la pénalisation si λ_k est une bonne estimation des multiplicateurs.

→ meilleur conditionnement

→ convergence plus rapide et précise que la méthode du critère augmenté

→ **méthode de lagrangien augmenté** appelée aussi **méthode des multiplicateurs**

Convergence

Pour ρ assez grand, **la solution x^* du problème initial est un minimum local du problème**

$$\min_{x \in \mathbb{R}^n} L_\rho(x, \lambda^*) = f(x) + \lambda^{*T} c(x) + \frac{1}{2} \rho \|c(x)\|^2 \rightarrow \text{pénalisation **exacte** si on connaît } \lambda^*$$

→ ne nécessite pas d'augmenter indéfiniment ρ pour obtenir la solution exacte x^*

3.4.3 Algorithme

- ❑ Algorithme de lagrangien augmenté
- ❑ Exemple

3.4.3 Algorithme

Méthode de lagrangien augmenté (ou méthode des multiplicateurs)

Les principales étapes d'une itération de lagrangien augmenté sont

- minimiser le lagrangien augmenté
- mettre à jour les paramètres de réglage

Minimisation du lagrangien augmenté

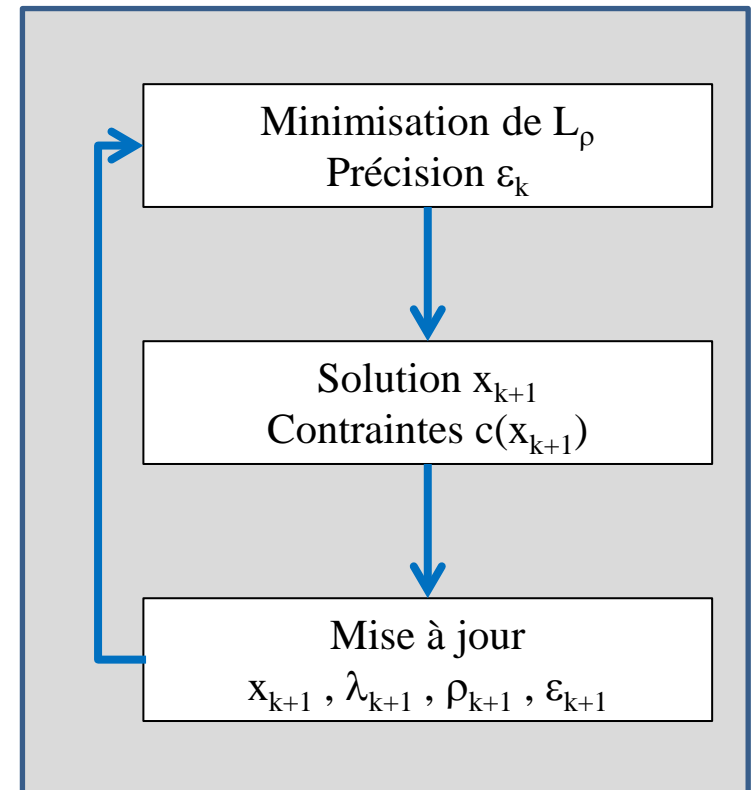
- Méthode de quasi-Newton
- Recherche linéaire ou région de confiance
- Précision d'arrêt sur gradient

Paramètres de réglage

- Multiplicateurs
- Pénalisation
- Précisions (gradient, contraintes)

Principales difficultés

- Précision contraintes → pénalisation forte
- Conditionnement → pénalisation faible
→ convergence précise difficile
→ réglages à adapter au cas par cas



3.4.3 Algorithme

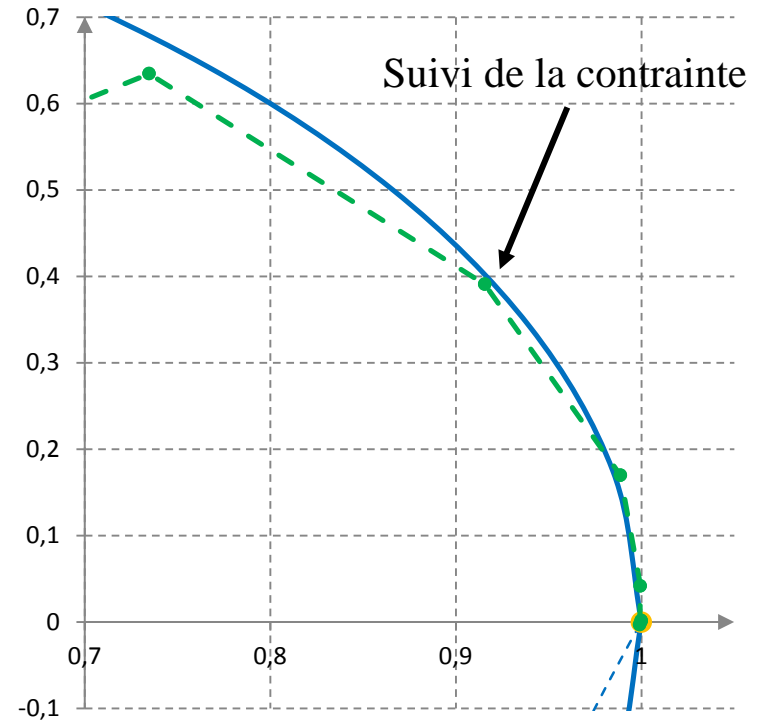
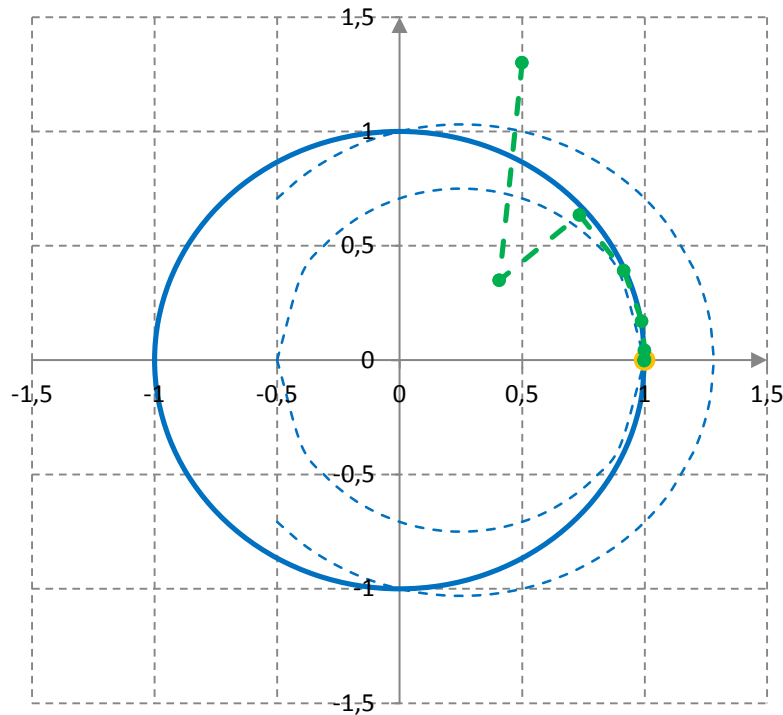
Méthode de lagrangien augmenté (ou méthode des multiplicateurs)

- Réglages à l'itération k :
 - multiplicateurs λ_k
 - pénalisation ρ_k
 - précision de résolution ε_k
 - précision sur les contraintes η_k
- Minimisation : $\min_{x \in \mathbb{R}^n} L_{\rho_k}(x, \lambda_k) \rightarrow \text{critère d'arrêt } \|\nabla_x L_{\rho_k}(x_{k+1}, \lambda_k)\| \leq \varepsilon_k$
 initialisation $x_k \rightarrow \text{solution } x_{k+1} \rightarrow \text{valeurs des contraintes} = c(x_{k+1})$
- Mise à jour des réglages à l'itération $k+1$ en fonction du respect des contraintes
 - Si $\|c(x_{k+1})\| < \eta_k$ \rightarrow **mise à jour des multiplicateurs** $\lambda_{k+1} = \lambda_k + \rho_k c(x_k)$
 (contraintes bien respectées) \rightarrow résolution plus précise $\varepsilon_{k+1} < \varepsilon_k$, $\eta_{k+1} < \eta_k$
 - Si $\|c(x_{k+1})\| > \eta_k$ \rightarrow **augmentation de la pénalisation** $\rho_{k+1} > \rho_k \rightarrow \times 10$
 (contraintes mal respectées) \rightarrow résolution moins précise $\varepsilon_{k+1} > \varepsilon_k$, $\eta_{k+1} > \eta_k$

3.4.3 Exemple

Exemple

- Minimisation de $f(x) = 2(x_1^2 + x_2^2 - 1) - x_1$ sous $c(x) = x_1^2 + x_2^2 - 1 = 0$
- Point initial : $x = \begin{pmatrix} 0.5 \\ 1.3 \end{pmatrix}$, $\lambda = 0$ Solution : $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\lambda = -\frac{3}{2}$

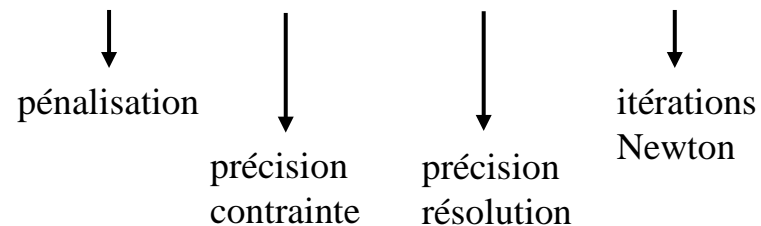


3.4.3 Exemple

Exemple

- Minimisation de $f(x) = 2(x_1^2 + x_2^2 - 1) - x_1$ sous $c(x) = x_1^2 + x_2^2 - 1 = 0$
- Point initial : $x = \begin{pmatrix} 0.5 \\ 1.3 \end{pmatrix}$, $\lambda = 0$ Solution : $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\lambda = -\frac{3}{2}$

Itération	x_1	x_2	λ	ρ	$c(x)$	$\ \nabla L_\rho(x, \lambda)\ $	Newton
1	0,50000	1,30000	0,00000	1	-0,71238	0,90050	1
2	0,40707	0,34917	-0,71238	10	-0,05788	0,90016	1
3	0,73467	0,63433	-1,29122	10	-0,00905	0,50091	2
4	0,91556	0,39077	-1,38175	10	0,00635	0,41807	2
5	0,98869	0,16985	-1,38175	100	0,00188	0,62061	2
6	0,99953	0,04158	-1,30283	100	-0,00188	0,01728	2
7	0,99905	-0,00320	-1,49103	100	-0,00009	0,00172	1
8	0,99995	0,00171	-1,50003	100	2,06E-06	0,00057	3
9	1,00000	0,00045	-1,50003	100	1,85E-06	0,00031	



Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
- 3. Optimisation avec contraintes**
 - 3.1 Simplexe
 - 3.2 Point intérieur
 - 3.3 Gradient projeté
 - 3.4 Lagrangien augmenté
 - 3.5 Programmation quadratique séquentielle**
 - 3.5.1 Equations KKT
 - 3.5.2 Modèle quadratique
 - 3.5.3 Globalisation
 - 3.5.4 Algorithme
 - 3.6 Convergence

3.5.1 Equations KKT

- ☐ Problème non linéaire
 - Conditions de minimum local
 - Equations KKT
- ☐ Méthode de Newton

3.5.1 Equations KKT

Problème avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \Leftrightarrow \min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \rightarrow m \text{ contraintes actives}$$

Conditions nécessaires de minimum local

- Lagrangien : $L(x, \lambda) = f(x) + \lambda^T c(x)$
- Gradient : $\nabla L(x, \lambda) = \begin{pmatrix} \nabla_x L(x, \lambda) \\ \nabla_\lambda L(x, \lambda) \end{pmatrix} = \begin{pmatrix} \nabla f(x) + \nabla c(x) \lambda \\ c(x) \end{pmatrix}$
- Hessien : $\nabla^2 L(x, \lambda) = \begin{pmatrix} \nabla_{xx}^2 L(x, \lambda) & \nabla_{x\lambda}^2 L(x, \lambda) \\ \nabla_{\lambda x}^2 L(x, \lambda) & \nabla_{\lambda\lambda}^2 L(x, \lambda) \end{pmatrix} = \begin{pmatrix} \nabla_{xx}^2 L(x, \lambda) & \nabla c(x) \\ \nabla c(x)^T & 0 \end{pmatrix}$
- Conditions d'ordre 1 : $\nabla L(x, \lambda) = \begin{pmatrix} \nabla_x L(x, \lambda) \\ \nabla_\lambda L(x, \lambda) \end{pmatrix} = 0 \rightarrow \text{équations KKT}$

On cherche à résoudre les équations KKT par la méthode de Newton.

3.5.1 Equations KKT

Equations KKT

Les conditions KKT donnent un système d'équations non linéaire en variables (x, λ)

$$\nabla L(x, \lambda) = \begin{pmatrix} \nabla_x L(x, \lambda) \\ \nabla_\lambda L(x, \lambda) \end{pmatrix} = 0$$

Méthode de Newton

Au point (x_k, λ_k) , l'itération de Newton donne un déplacement d vérifiant

$$\nabla^2 L(x_k, \lambda_k) d = -\nabla L(x_k, \lambda_k) \quad \text{avec } d = \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} \in \mathbb{R}^{n+m}$$

$$\Leftrightarrow \begin{pmatrix} \nabla_{xx}^2 L(x_k, \lambda_k) & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) \lambda_k \\ c(x_k) \end{pmatrix}$$

$$\Leftrightarrow \boxed{\begin{cases} \nabla_{xx}^2 L(x_k, \lambda_k) d_x + \nabla c(x_k) d_\lambda = -\nabla_x L(x_k, \lambda_k) \\ \nabla c(x_k)^T d_x = -c(x_k) \end{cases}} \rightarrow \text{équations de Newton}$$

On peut écrire un **problème quadratique-linéaire** en variables (d_x, d_λ)
 dont les conditions d'optimalité sont les équations de Newton au point (x_k, λ_k) .

3.5.2 Modèle quadratique

- ☐ Problème quadratique local
- ☐ Equivalence avec la méthode de Newton
- ☐ Interprétation
- ☐ Formulation simplifiée
- ☐ Résolution

3.5.2 Modèle quadratique

Problème quadratique local

Au point (x_k, λ_k) , on considère le problème d'optimisation sur la variable $d_{QP} \in \mathbb{R}^n$

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla_x L(x_k, \lambda_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} \quad \text{sous} \quad \nabla c(x_k)^T d_{QP} + c(x_k) = 0$$

→ problème quadratique-linéaire noté (QP)

- Lagrangien L_{QP} avec des multiplicateurs λ_{QP} sur les contraintes : $\nabla c(x_k)^T d_{QP} + c(x_k) = 0$

$$L_{QP}(d_{QP}, \lambda_{QP}) = \nabla_x L(x_k, \lambda_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} + \lambda_{QP}^T (\nabla c(x_k)^T d_{QP} + c(x_k))$$

- Conditions d'ordre 1 : $\nabla L_{QP}(d_{QP}, \lambda_{QP}) = \begin{pmatrix} \nabla_d L_{QP}(d_{QP}, \lambda_{QP}) \\ \nabla_\lambda L_{QP}(d_{QP}, \lambda_{QP}) \end{pmatrix} = 0$

$$\Leftrightarrow \begin{cases} \nabla_x L(x_k, \lambda_k) + \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} + \nabla c(x_k) \lambda_{QP} = 0 \\ \nabla c(x_k)^T d_{QP} + c(x_k) = 0 \end{cases}$$

→ système linéaire sur les variables primales $d_{QP} \in \mathbb{R}^n$ et les variables duales $\lambda_{QP} \in \mathbb{R}^m$

3.5.2 Modèle quadratique

Equivalence avec la méthode de Newton

On se place au point (x_k, λ_k) .

- L'itération de Newton pour résoudre les conditions KKT donne le système en variables (d_x, d_λ)

$$\begin{cases} \nabla_{xx}^2 L(x_k, \lambda_k) d_x + \nabla c(x_k) d_\lambda = -\nabla_x L(x_k, \lambda_k) \\ \nabla c(x_k)^T d_x = -c(x_k) \end{cases}$$

- Le problème quadratique-linéaire (QP) en variables $d_{QP} \in \mathbb{R}^n$

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla_x L(x_k, \lambda_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} \quad \text{sous} \quad \nabla c(x_k)^T d_{QP} + c(x_k) = 0$$

a pour conditions d'ordre 1 le système en variables (d_{QP}, λ_{QP})

$$\begin{cases} \nabla_x L(x_k, \lambda_k) + \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} + \nabla c(x_k) \lambda_{QP} = 0 \\ \nabla c(x_k)^T d_{QP} + c(x_k) = 0 \end{cases}$$

- **Les 2 systèmes linéaires sont identiques** en posant : $\begin{cases} d_{QP} = d_x \in \mathbb{R}^n \\ \lambda_{QP} = d_\lambda \in \mathbb{R}^m \end{cases}$

3.5.2 Modèle quadratique

Interprétation

- L'itération de Newton au point (x_k, λ_k) équivaut à la résolution du problème quadratique local

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla_x L(x_k, \lambda_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} \quad \text{sous} \quad \nabla c(x_k)^T d_{QP} + c(x_k) = 0 \rightarrow \begin{cases} d_x = d_{QP} \\ d_\lambda = \lambda_{QP} \end{cases}$$

→ multiplicateur λ_{QP}

- On peut formuler le problème quadratique local (QP) en

$$\min_{d_{QP} \in \mathbb{R}^n} \tilde{L}(d_{QP}) \quad \text{sous} \quad \tilde{c}(d_{QP}) = 0 \quad \text{avec} \quad \begin{cases} \tilde{L}(d_{QP}) = L(x_k + d_{QP}, \lambda_k) + o(\|d_{QP}\|^2) & \rightarrow L \text{ à l'ordre 2 en } x \\ \tilde{c}(d_{QP}) = c(x_k + d_{QP}) + o(\|d_{QP}\|) & \rightarrow c \text{ à l'ordre 1 en } x \end{cases}$$

→ minimisation d'un modèle quadratique du lagrangien
 sous un modèle linéaire des contraintes

Récapitulatif

- Optimisation sans contrainte
 Newton \Leftrightarrow Minimiser un modèle quadratique de la fonction
- Optimisation avec contrainte
 Newton \Leftrightarrow Minimiser un modèle quadratique du lagrangien
 sous un modèle linéaire des contraintes

3.5.2 Modèle quadratique

Formulation simplifiée

- L'itération de Newton au point (x_k, λ_k) s'écrit en variables (d_x, d_λ) :

$$\begin{cases} \nabla_{xx}^2 L(x_k, \lambda_k) d_x + \nabla c(x_k) d_\lambda = -\nabla_x L(x_k, \lambda_k) = -\nabla f(x_k) - \nabla c(x_k) \lambda_k \\ \nabla c(x_k)^T d_x = -c(x_k) \end{cases}$$

- En faisant le changement de variable $d_l = \lambda_k + d_\lambda$, on obtient

$$\begin{cases} \nabla_{xx}^2 L(x_k, \lambda_k) d_x + \nabla c(x_k) d_l = -\nabla f(x_k) \\ \nabla c(x_k)^T d_x = -c(x_k) \end{cases}$$

- L'itération de Newton en variables (d_x, d_l) équivaut au problème quadratique local

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla f(x_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} \quad \text{sous} \quad \nabla c(x_k)^T d_{QP} + c(x_k) = 0$$

→ multiplicateur λ_{QP}

$$\begin{aligned} &\rightarrow \begin{cases} d_x = d_{QP} \\ d_l = \lambda_{QP} \end{cases} \\ &\rightarrow \begin{cases} d_x = d_{QP} \\ d_\lambda = \lambda_{QP} - \lambda_k \end{cases} \end{aligned}$$

→ même formulation avec $\nabla f(x_k)$ au lieu de $\nabla_x L(x_k, \lambda_k)$

3.5.2 Modèle quadratique

Solution du problème quadratique

- On peut calculer explicitement la solution du problème quadratique local

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla f(x_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} \quad \text{sous} \quad \nabla c(x_k)^T d_{QP} + c(x_k) = 0$$

$$\Leftrightarrow \min_{d_{QP} \in \mathbb{R}^n} \frac{1}{2} d_{QP}^T Q d_{QP} + c^T d_{QP} \quad \text{sous} \quad A d_{QP} = b \quad \text{avec} \quad \begin{cases} Q = \nabla_{xx}^2 L(x_k, \lambda_k) \\ c = \nabla f(x_k) \end{cases} \quad \text{et} \quad \begin{cases} A = \nabla c(x_k)^T \\ b = -c(x_k) \end{cases}$$

- Si Q est définie positive, la solution est :
$$\begin{cases} \lambda_{QP} = (A Q^{-1} A^T)^{-1} (A Q^{-1} c + b) \\ d_{QP} = Q^{-1} (A^T \lambda_{QP} - c) \end{cases}$$
- Si Q n'est pas définie positive, on remplace Q par une matrice H définie positive « proche »
 → factorisation de Cholesky modifiée
 → ou $H = Q + \tau I$ avec τ suffisamment grand
- Le déplacement à partir du point courant (x_k, λ_k) est :
$$\begin{cases} d_x = d_{QP} \\ d_\lambda = \lambda_{QP} - \lambda_k \end{cases} \rightarrow \begin{cases} x_{k+1} = x_k + d_{QP} \\ \lambda_{k+1} = \lambda_{QP} \end{cases}$$
- La résolution explicite est généralement trop coûteuse à cause des inversions de matrices.
 On résout le problème quadratique-linéaire par un algorithme itératif dédié.

3.5.3 Globalisation

- ☐ Méthode de Newton
- ☐ SQP avec recherche linéaire
- ☐ SQP avec région de confiance
- ☐ Correction d'ordre 2

3.5.3 Globalisation

Méthode de Newton

- La résolution des équations KKT par la méthode de Newton équivaut à la résolution d'une suite de problèmes quadratiques
→ **programmation quadratique séquentielle (SQP)**
- La méthode de Newton n'est pas globalement convergente.
→ globalisation nécessaire par recherche linéaire ou région de confiance
→ fonction mérite F mesurant la progression de l'algorithme vis-à-vis
 - de l'amélioration du critère
 - du respect des contraintes → critère augmenté, lagrangien augmenté

Globalisation

Deux grandes méthodes de globalisation

- **Recherche linéaire**
La solution du problème QP sert de direction de descente pour une recherche linéaire.
→ réglage du pas de déplacement pour améliorer la fonction mérite
- **Région de confiance**
On ajoute au problème QP une contrainte de région de confiance.
→ réglage du rayon de confiance en fonction du rapport de réduction de la fonction mérite

3.5.3 SQP avec recherche linéaire

Principe

- On effectue une **recherche linéaire suivant la direction \mathbf{d}_x** de la solution du problème QP.
- L'amélioration est mesurée par une fonction mérite F.
 La fonction F doit prendre en compte l'amélioration du critère et le respect des contraintes.
 → pénalisation de la violation des contraintes
 → différents choix possibles
- La fonction F doit permettre de converger vers l'optimum du problème initial.

Fonction de mérite exacte

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases} \rightarrow \text{problème (PO)}$$

F est une **fonction de mérite exacte** si tout minimum local x^* de (PO)
 est aussi un minimum local sans contrainte de F

$$x^* \text{ minimum local de } \min_{x \in \mathbb{R}^n} f(x) \text{ sous } \begin{cases} c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$$

$$\Rightarrow x^* \text{ minimum local de } \min_{x \in \mathbb{R}^n} F(x)$$

3.5.3 SQP avec recherche linéaire

Fonction mérite I_1

- Problème avec contraintes égalité

$$\min_{x \in \mathbb{R}^n} f(x) \text{ sous } c(x) = 0 \quad \rightarrow \text{contraintes actives}$$

- Critère augmenté avec la norme 1 : $F_{1,\rho}(x) = f(x) + \rho \|c(x)\|_1$ avec $\|c(x)\|_1 = \sum_{i=1}^m |c_i(x)|$

$F_{1,\rho}$ est une fonction de mérite exacte lorsque le coefficient de pénalisation ρ est assez grand

Théorème

On suppose que (x^*, λ^*) vérifient les conditions suffisantes d'optimalité du problème (PO)

$$\begin{cases} \nabla L(x^*, \lambda^*) = 0 \\ d^T \nabla_{xx}^2 L(x^*, \lambda^*) d \geq 0, \forall d \in D(x^*) \end{cases} \quad \text{avec } D(x^*) = \text{cône des directions en } x^*$$

$$F_{1,\rho} \text{ est une fonction de mérite exacte si } \rho > \max_{i=1, \dots, m} |\lambda_i^*|$$

Réglage de ρ

- ρ trop grand \rightarrow mauvais conditionnement de la fonction mérite $F_{1,\rho}$
difficulté de convergence
- ρ trop petit \rightarrow respect insuffisant des contraintes
optimum \neq problème initial (fonction mérite $F_{1,\rho}$ non exacte)

3.5.3 SQP avec recherche linéaire

Dérivée directionnelle de $F_{1,\rho}$

- La direction de recherche linéaire est donnée par la solution d_x du problème QP

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla f(x_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} \quad \text{sous} \quad \nabla c(x_k)^T d_{QP} + c(x_k) = 0 \quad \rightarrow \quad d_x = d_{QP}$$

- $F_{1,\rho}$ n'est pas dérivable \rightarrow évaluation de la **dérivée directionnelle** suivant la direction d_x

$$F_{1,\rho}(x) = f(x) + \rho \|c(x)\|_1 \rightarrow (F_{1,\rho})'_{d_x} = \left(\frac{dF_{1,\rho}(x_k + s d_x)}{ds} \right)_{s=0} = \nabla f(x_k)^T d_x - \rho \|c(x_k)\|_1$$

- $(F_{1,\rho})'_{d_x}$ vérifie $(F_{1,\rho})'_{d_x} \leq -d_x^T \nabla_{xx}^2 L(x_k, \lambda_k) d_x - (\rho - \|d_1\|_\infty) \|c(x)\|_1$ avec $d_1 = \lambda_k + d_\lambda$

Valeur minimale de ρ

- d_x est une **direction de descente** pour $F_{1,\rho}$ si $\rho > \|d_1\|_\infty - \frac{d_x^T \nabla_{xx}^2 L(x_k, \lambda_k) d_x}{\|c(x)\|_1}$

- Condition suffisante $\rho > \|d_1\|_\infty$ si $\nabla_{xx}^2 L(x_k, \lambda_k) > 0$

- $F_{1,\rho}$ est une fonction de mérite exacte si $\boxed{\rho > \max_{i=1, \dots, m} |\lambda_i^*|}$ \rightarrow conditions cohérentes
 car $d_1 = \lambda_k + d_\lambda$ avec $d_\lambda^* = 0$

3.5.3 SQP avec région de confiance

Principe

- On ajoute au problème quadratique local une **contrainte de région de confiance**.

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla f(x_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} \quad \text{sous} \quad \begin{cases} \nabla c(x_k)^T d_{QP} + c(x_k) = 0 \\ \|d_{QP}\| \leq r \end{cases}$$

- On peut exprimer la contrainte de région de confiance en norme 2 ou en norme ∞ .
- L'amélioration est mesurée par une fonction mérite F.
 Le rayon de confiance est réglé en fonction du rapport de réduction de la fonction mérite.

Fonction de mérite l_2

- On peut prendre une fonction mérite avec pénalisation quadratique

$$F_{2,\rho}(x) = f(x) + \rho \|c(x)\|_2 \quad \text{avec} \quad \|c(x)\|_2 = \sqrt{\sum_{i=1}^m c_i(x)^2}$$

- La fonction modèle quadratique correspondante est

$$\hat{F}_{2,\rho}(d_{QP}) = \nabla f(x_k)^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L(x_k, \lambda_k) d_{QP} + \rho \|\nabla c(x_k)^T d_{QP} + c(x_k)\|_2$$

- Le rapport de réduction est défini par
$$\frac{F_{2,\rho}(x_k) - F_{2,\rho}(x_k + d_{PQ})}{\hat{F}_{2,\rho}(x_k) - \hat{F}_{2,\rho}(x_k + d_{PQ})}$$

3.5.3 Correction d'ordre 2

Effet Maratos

- La direction de déplacement obtenue en résolvant le problème QP peut simultanément :
 - rapprocher de l'optimum
 - dégrader la fonction mérite basée sur l'amélioration du critère et des contraintes
- Ceci peut conduire à bloquer un algorithme SQP en rejetant des déplacements favorables.

Correction d'ordre 2

- On résout **une 1^{ère} fois** le problème QP au point $x_k \rightarrow$ solution d_{QP}

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla f_k^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L_k d_{QP} \quad \text{sous} \quad \nabla c_k^T d_{QP} + c_k = 0$$

- On évalue la valeur des contraintes au point $x_k + d_{QP} \rightarrow \boxed{c(x_k + d_{QP}) = \delta c}$
- Si la valeur de $c(x_k + d_{QP})$ est trop différente de 0 (non-linéarité des contraintes), on résout **une 2^{ème} fois** le problème QP au point x_k **en corrigeant la valeur des contraintes**

$$\min_{d_{QP} \in \mathbb{R}^n} \nabla f_k^T d_{QP} + \frac{1}{2} d_{QP}^T \nabla_{xx}^2 L_k d_{QP} \quad \text{sous} \quad \nabla c_k^T d_{QP} + c_k = -\delta c \quad \rightarrow \text{correction d'ordre 2}$$

- On obtient une direction de déplacement qui tient compte de la non-linéarité des contraintes.

3.5.3 Exemple

Effet Maratos

- Minimisation de $f(x) = 2(x_1^2 + x_2^2 - 1) - x_1$ sous $c(x) = x_1^2 + x_2^2 - 1 = 0$

- Solution : $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\lambda^* = -\frac{3}{2} \rightarrow \nabla_{xx}^2 L(x^*, \lambda^*) = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Point courant : $x_k = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \rightarrow \nabla f = \begin{pmatrix} 4 \cos \theta - 1 \\ 4 \sin \theta \end{pmatrix}$, $\nabla c = \begin{pmatrix} 2 \cos \theta \\ 2 \sin \theta \end{pmatrix}$

- Problème QP : $\min_{d_1, d_2} (4 \cos \theta - 1)d_1 + 4d_2 \sin \theta + \frac{1}{2}(d_1^2 + d_2^2)$ sous $d_1 \cos \theta + d_2 \sin \theta = 0$

en prenant $\nabla_{xx}^2 L = I$

- Solution QP : $d_1 = -d_2 \tan \theta \rightarrow \min_{d_2} d_2 \tan \theta + \frac{1}{2 \cos^2 \theta} d_2^2 \rightarrow d_2 = -\sin \theta \cos \theta$

$$\rightarrow d_k = \begin{pmatrix} \sin^2 \theta \\ -\sin \theta \cos \theta \end{pmatrix} = \sin \theta \begin{pmatrix} \sin \theta \\ -\cos \theta \end{pmatrix}$$

- Nouveau point : $x_{k+1} = \begin{pmatrix} \cos \theta + \sin^2 \theta \\ \sin \theta - \sin \theta \cos \theta \end{pmatrix}$

3.5.3 Exemple

Effet Maratos

- Minimisation de $f(x) = 2(x_1^2 + x_2^2 - 1) - x_1$ sous $c(x) = x_1^2 + x_2^2 - 1 = 0$

On évalue en $x_k = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}$ et $x_{k+1} = \begin{pmatrix} \cos \theta + \sin^2 \theta \\ \sin \theta - \sin \theta \cos \theta \end{pmatrix}$

- La contrainte :
$$\begin{cases} c(x_k) = \cos^2 \theta + \sin^2 \theta - 1 \\ c(x_{k+1}) = (\cos \theta + \sin^2 \theta)^2 + (\sin \theta - \sin \theta \cos \theta)^2 - 1 \end{cases} \Rightarrow \begin{cases} c(x_k) = 0 \\ c(x_{k+1}) = \sin^2 \theta \end{cases}$$
- Le critère :
$$\begin{cases} f(x_k) = 2c(x_k) - \cos \theta \\ f(x_{k+1}) = 2c(x_{k+1}) - (\cos \theta + \sin^2 \theta) \end{cases} \Rightarrow \begin{cases} f(x_k) = -\cos \theta \\ f(x_{k+1}) = \sin^2 \theta - \cos \theta \end{cases}$$
- L'écart à l'optimum $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$:
$$\begin{cases} x_k - x^* = \begin{pmatrix} \cos \theta - 1 \\ \sin \theta \end{pmatrix} \\ x_{k+1} - x^* = (1 - \cos \theta) \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \end{cases} \Rightarrow \begin{aligned} \|x_k - x^*\|^2 &= 2(1 - \cos \theta) \\ \|x_{k+1} - x^*\|^2 &= 1 - \cos \theta \end{aligned}$$

Le point x_{k+1} est **plus proche de l'optimum** que le point x_k , alors que :

- la valeur du critère est dégradée : $f(x_{k+1}) > f(x_k)$
 - la valeur de la contrainte est dégradée : $c(x_{k+1}) \neq 0$
- déplacement rejeté

3.5.3 Exemple

Correction d'ordre 2

- Minimisation de $f(x) = 2(x_1^2 + x_2^2 - 1) - x_1$ sous $c(x) = x_1^2 + x_2^2 - 1 = 0$

On évalue la contrainte au point $x_{k+1} = \begin{pmatrix} \cos \theta + \sin^2 \theta \\ \sin \theta - \sin \theta \cos \theta \end{pmatrix} \rightarrow c(x_{k+1}) = \sin^2 \theta = \delta c$

- Le problème QP est résolu **une 2^{ème} fois en corrigeant la valeur de la contrainte** :

$$\min_{d_1, d_2} (4 \cos \theta - 1)d_1 + 4d_2 \sin \theta + \frac{1}{2}(d_1^2 + d_2^2) \text{ sous } d_1 \cos \theta + d_2 \sin \theta = -\sin^2 \theta$$

- Solution QP :

$$d_1 = -(d_2 + \sin \theta) \tan \theta \rightarrow \min_{d_2} d_2 \tan \theta + \frac{1}{2}d_2^2 + \frac{1}{2\cos^2 \theta}(d_2 + \sin \theta)^2 \rightarrow d_2 = -\sin \theta \cos \theta - \sin^3 \theta$$

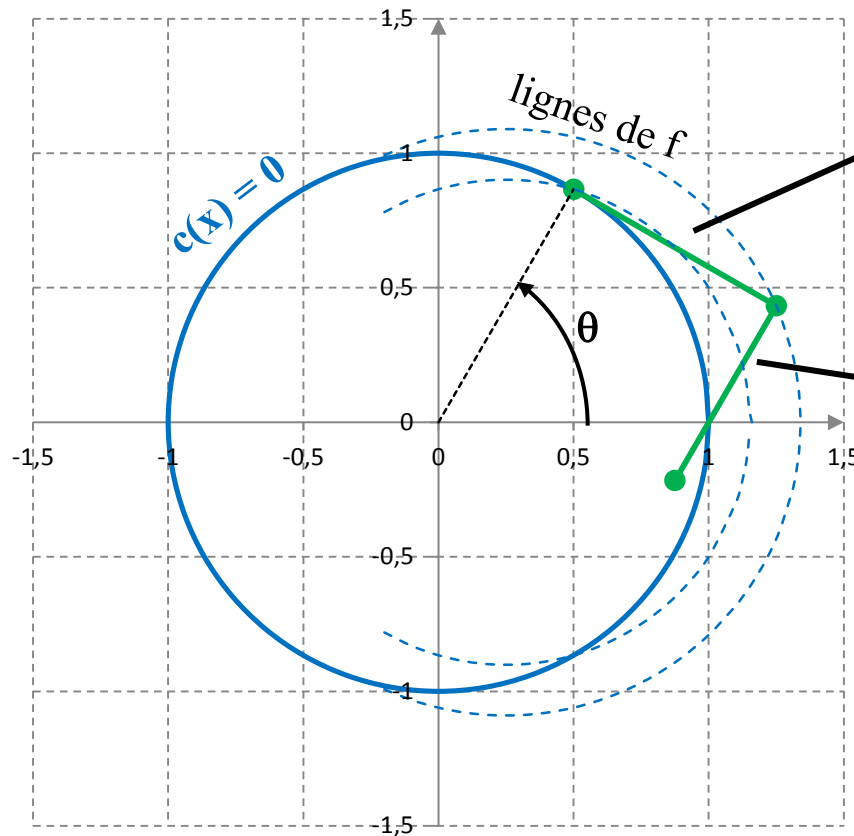
$$\rightarrow d_k = \sin \theta \begin{pmatrix} \sin \theta \\ -\cos \theta \end{pmatrix} - \sin^2 \theta \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \rightarrow \text{correction d'ordre 2}$$

- Déplacement d'ordre 1 : **tangent** de norme $\sin \theta \rightarrow$ suppose la contrainte linéaire
- Déplacement d'ordre 2 : **radial** de norme $\sin^2 \theta \rightarrow$ corrige la non-linéarité

3.5.3 Exemple

Correction d'ordre 2

- Minimisation de $f(x) = 2(x_1^2 + x_2^2 - 1) - x_1$ sous $c(x) = x_1^2 + x_2^2 - 1 = 0$



Ordre 1 $\rightarrow d = \sin \theta \begin{pmatrix} \sin \theta \\ -\cos \theta \end{pmatrix}$

Ordre 2 $\rightarrow d = -\sin^2 \theta \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}$

3.5.4 Algorithme

□ Algorithme SQP

□ Exemples

- Exemple 1 : comparaison Newton – SQP
- Exemple 2 : SQP sans – avec globalisation
- Exemple 3 : comparaison SQP – Lagrangien augmenté

3.5.4 Algorithme

Algorithme SQP

Les principales étapes d'une itération SQP sont

- construire le modèle quadratique local au point courant
- résoudre le problème quadratique-linéaire
- appliquer une technique de globalisation

Modèle quadratique local

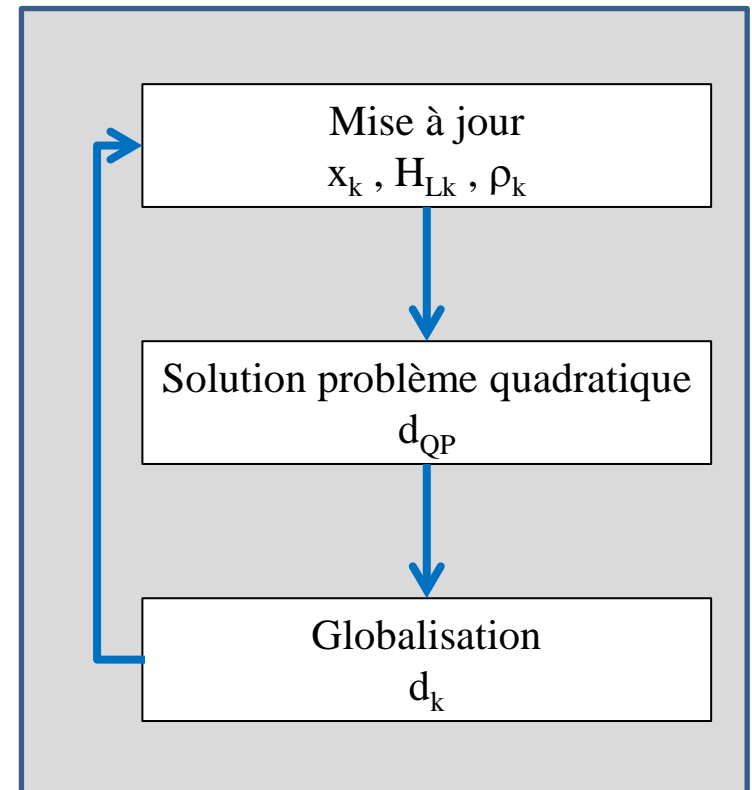
- Sélection des contraintes actives
- Mise à jour du hessien (quasi-Newton)

Résolution du problème quadratique-linéaire

- Modification pour avoir un hessien défini positif
- Algorithme spécialisé problème quadratique

Globalisation

- Mise à jour de la pénalisation de la fonction mérite
- Recherche linéaire ou région de confiance
 - grand nombre de variantes possibles
 - stratégies et réglages à adapter au cas par cas



3.5.4 Exemples

Exemple 1 : Comparaison Newton – SQP

- Minimisation de $f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$ sous $c(x) = 2x_1^2 - x_2 = 0$
- Comparaison de 3 méthodes** : résolution exacte , méthode de Newton , algorithme SQP

Résolution exacte

- Lagrangien : $L(x, \lambda) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 + \lambda(2x_1^2 - x_2)$
- Conditions KKT :
$$\begin{cases} 4x_1 - 2x_2 - 4 + 4\lambda x_1 = 0 \\ 4x_2 - 2x_1 - 6 - \lambda = 0 \\ 2x_1^2 - x_2 = 0 \end{cases} \rightarrow \text{solution} \begin{cases} x_1^* \approx 1.06902 \\ x_2^* \approx 2.28563 \\ \lambda^* \approx 1.00446 \end{cases}$$

Matrices utiles

- Newton : $F(x_1, x_2, \lambda) = \begin{pmatrix} 4x_1 - 2x_2 - 4 + 4\lambda x_1 \\ 4x_2 - 2x_1 - 6 - \lambda \\ 2x_1^2 - x_2 \end{pmatrix} \Rightarrow \nabla F(x_1, x_2, \lambda) = \begin{pmatrix} 4 + 4\lambda & -2 & 4x_1 \\ -2 & 4 & -1 \\ 4x_1 & -1 & 0 \end{pmatrix}$
- SQP : $\nabla f(x) = \begin{pmatrix} 4x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 - 6 \end{pmatrix} \quad \nabla c(x) = \begin{pmatrix} 4x_1 \\ -1 \end{pmatrix} \quad \nabla_{xx}^2 L(x, \lambda) = \begin{pmatrix} 4 + 4\lambda & -2 \\ -2 & 4 \end{pmatrix}$

3.5.4 Exemples

Exemple 1 : Méthode de Newton

- **Itération de Newton :**

$$\nabla F(x_1, x_2, \lambda) \begin{pmatrix} d_{x1} \\ d_{x2} \\ d_\lambda \end{pmatrix} = -F(x_1, x_2, \lambda)$$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \rightarrow F(x_1, x_2, \lambda) = \begin{pmatrix} -6 \\ -2 \\ -1 \end{pmatrix} \quad \nabla F(x_1, x_2, \lambda) = \begin{pmatrix} 4 & -2 & 0 \\ -2 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$

- Itération 1 : $\begin{pmatrix} 4 & -2 & 0 \\ -2 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} d_{x1} \\ d_{x2} \\ d_\lambda \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} d_{x1} \\ d_{x2} \\ d_\lambda \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ -8 \end{pmatrix}$

- Nouveau point : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} \rightarrow \begin{pmatrix} x_1 + d_{x1} \\ x_2 + d_{x2} \\ \lambda + d_\lambda \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -8 \end{pmatrix}$

- La méthode de Newton converge en **7 itérations**, sans globalisation.

3.5.4 Exemples

Exemple 1 : Méthode de Newton

Itération	x_1	x_2	λ	$f(x_1, x_2)$	$c(x_1, x_2)$	$F(x_1, x_2, \lambda)$	$dF(x_1, x_2, \lambda)$		
1	0,00000	1,00000	0,00000	-4,00000	-1,00000	-6,000	4,000	-2,000	0,000
						-2,000	-2,000	4,000	-1,000
						-1,000	0,000	-1,000	0,000
2	1,00000	0,00000	-8,00000	-2,00000	2,00000	-32,000	-28,000	-2,000	4,000
						0,000	-2,000	4,000	-1,000
						2,000	4,000	-1,000	0,000
3	1,20000	2,80000	2,80000	-9,76000	0,08000	8,640	15,200	-2,000	4,800
						0,000	-2,000	4,000	-1,000
						0,080	4,800	-1,000	0,000
4	1,08639	2,33466	1,16588	-10,16445	0,02582	7,4E-01	8,664	-2,000	4,346
						0,0E+00	-2,000	4,000	-1,000
						2,6E-02	4,346	-1,000	0,000
5	1,06933	2,28636	1,00676	-10,14342	0,00058	1,1E-02	8,027	-2,000	4,277
						0,0E+00	-2,000	4,000	-1,000
						5,8E-04	4,277	-1,000	0,000
6	1,06902	2,28563	1,00446	-10,14283	1,88E-07	2,8E-06	8,018	-2,000	4,276
						0,0E+00	-2,000	4,000	-1,000
						1,9E-07	4,276	-1,000	0,000
7	1,06902	2,28563	1,00446	-10,14283	1,60E-14	2,1E-13	8,018	-2,000	4,276
						0,0E+00	-2,000	4,000	-1,000
						1,6E-14	4,276	-1,000	0,000

3.5.4 Exemples

Exemple 1 : Algorithme SQP

- **Problème quadratique linéaire :**

$$\min_{\mathbf{d}_x \in \mathbb{R}^2} \nabla f(\mathbf{x})^T \mathbf{d}_x + \frac{1}{2} \mathbf{d}_x^T \nabla_{xx}^2 L(\mathbf{x}, \lambda) \mathbf{d}_x \text{ sous } \nabla c(\mathbf{x})^T \mathbf{d}_x + c(\mathbf{x}) = 0$$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \rightarrow \nabla f(\mathbf{x}) = \begin{pmatrix} -6 \\ -2 \end{pmatrix} \quad \nabla c(\mathbf{x}) = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \quad \nabla_{xx}^2 L(\mathbf{x}, \lambda) = \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix}$

- Itération 1 : $\min_{\mathbf{d}_{x1}, \mathbf{d}_{x2}} \begin{pmatrix} -6 \\ -2 \end{pmatrix}^T \begin{pmatrix} \mathbf{d}_{x1} \\ \mathbf{d}_{x2} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \mathbf{d}_{x1} \\ \mathbf{d}_{x2} \end{pmatrix}^T \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{x1} \\ \mathbf{d}_{x2} \end{pmatrix} \text{ sous } \begin{pmatrix} 0 \\ -1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{x1} \\ \mathbf{d}_{x2} \end{pmatrix}_x - 1 = 0$

$$\Rightarrow \begin{cases} \min_{\mathbf{d}_{x1}} 2\mathbf{d}_{x1}^2 - 4\mathbf{d}_{x1} \\ \mathbf{d}_{x2} = -1 \end{cases} \Rightarrow \begin{cases} \mathbf{d}_{x1} = 1 \\ \mathbf{d}_{x2} = -1 \end{cases} \text{ avec } \lambda_{QP} = -8$$

- Nouveau point : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} \rightarrow \begin{pmatrix} x_1 + \mathbf{d}_{x1} \\ x_2 + \mathbf{d}_{x2} \\ \lambda_{QP} \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -8 \end{pmatrix}$

- On retrouve les itérations de la méthode de Newton sans globalisation.
 Le multiplicateur λ est directement le multiplicateur λ_{QP} du problème quadratique.

3.5.4 Exemples

Exemple 1 : Algorithme SQP

Itération	x_1	x_2	λ	$f(x_1, x_2)$	$c(x_1, x_2)$	∇f	∇c	∇L	$\nabla^2 L(x_1, x_2, \lambda)$	
1	0,00000	1,00000	0,00000	-4,00000	-1,00000	-6,000 -2,000	0,000 -1,000	-6,000 -2,000	4,000 -2,000	-2,000 4,000
2	1,00000	0,00000	-8,00000	-2,00000	2,00000	0,000 -8,000	4,000 -1,000	-32,000 0,000	-28,000 -2,000	-2,000 4,000
3	1,20000	2,80000	2,80000	-9,76000	0,08000	-4,800 2,800	4,800 -1,000	8,640 0,000	15,200 -2,000	-2,000 4,000
4	1,08639	2,33466	1,16588	-10,16445	0,02582	-4,324 1,166	4,346 -1,000	0,74262 0,00000	8,664 -2,000	-2,000 4,000
5	1,06933	2,28636	1,00676	-10,14342	0,00058	-4,295 1,007	4,277 -1,000	1,1E-02 0,0E+00	8,027 -2,000	-2,000 4,000
6	1,06902	2,28563	1,00446	-10,14283	1,9E-07	-4,295 1,004	4,276 -1,000	2,8E-06 0,0E+00	8,018 -2,000	-2,000 4,000
7	1,06902	2,28563	1,00446	-10,14283	1,6E-14	-4,295 1,004	4,276 -1,000	2,1E-13 0,0E+00	8,018 -2,000	-2,000 4,000

3.5.4 Exemples

Exemple 2 : SQP sans – avec globalisation

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$
- Différents points initiaux \rightarrow **globalisation nécessaire** pour assurer la convergence

Résolution exacte

- Lagrangien : $L(x, \lambda) = x_1 + x_2 + \lambda(x_1^2 + (x_2 - 1)^2 - 1)$
- Conditions KKT :
$$\begin{cases} 1 + 2\lambda x_1 = 0 \\ 1 + 2\lambda(x_2 - 1) = 0 \\ x_1^2 + (x_2 - 1)^2 - 1 = 0 \end{cases} \rightarrow \text{solution} \begin{cases} x_1^* = -1/\sqrt{2} \approx -0.70711 \\ x_2^* = 1 - 1/\sqrt{2} \approx 0.29289 \\ \lambda^* = 1/\sqrt{2} \approx 0.70711 \end{cases}$$

Matrices utiles

- Gradient : $\nabla f(x) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \nabla c(x) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 1) \end{pmatrix} \quad \nabla_{xx}^2 L(x, \lambda) = \begin{pmatrix} 2\lambda & 0 \\ 0 & 2\lambda \end{pmatrix}$
- Modification du hessien** : $H = \nabla_{xx}^2 L(x, \lambda) + \tau I = (2\lambda + \tau) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ avec $2\lambda + \tau > 0$
- Recherche linéaire** : pas s suivant la direction d_x (= solution du problème QP)

3.5.4 Exemples

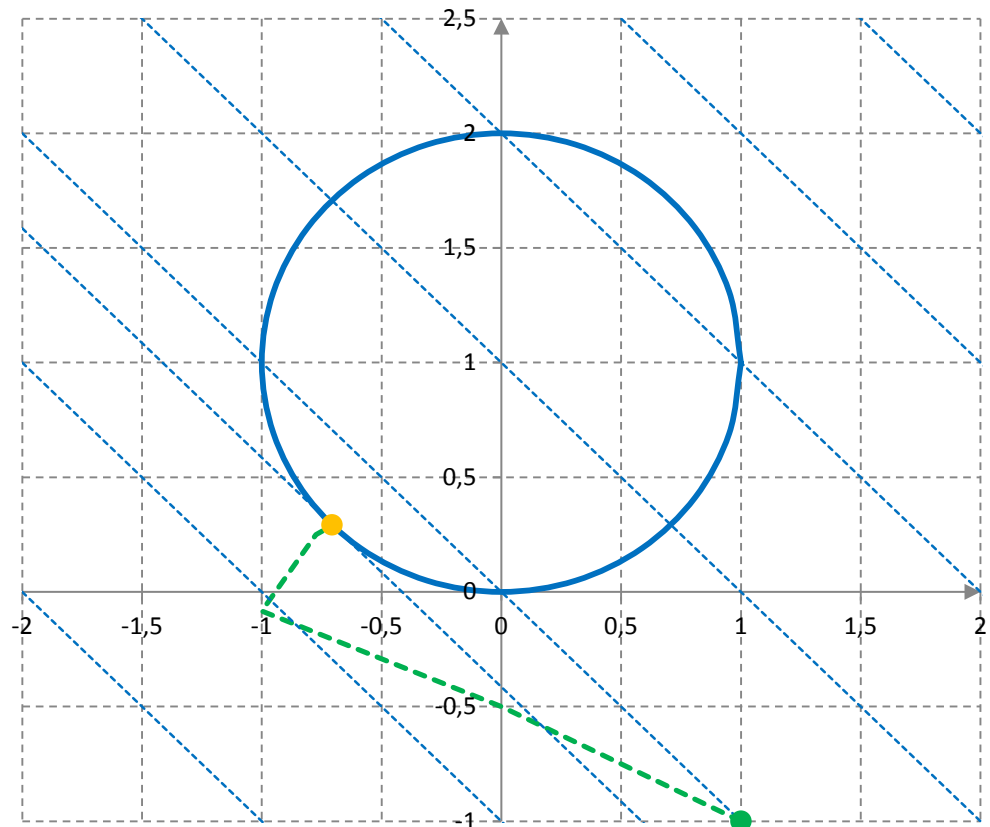
Exemple 2 : SQP sans globalisation

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$

- Algorithme SQP sans globalisation
 → convergence rapide

Itération	x1	x2	λ
1	1,00000	-1,00000	1,00000
2	0,00000	-0,50000	0,50000
3	-1,00000	-0,08333	0,47222
4	-0,77401	0,24973	0,60672
5	-0,70743	0,28900	0,69818
6	-0,70714	0,29291	0,70707
7	-0,70711	0,29289	0,70711



3.5.4 Exemples

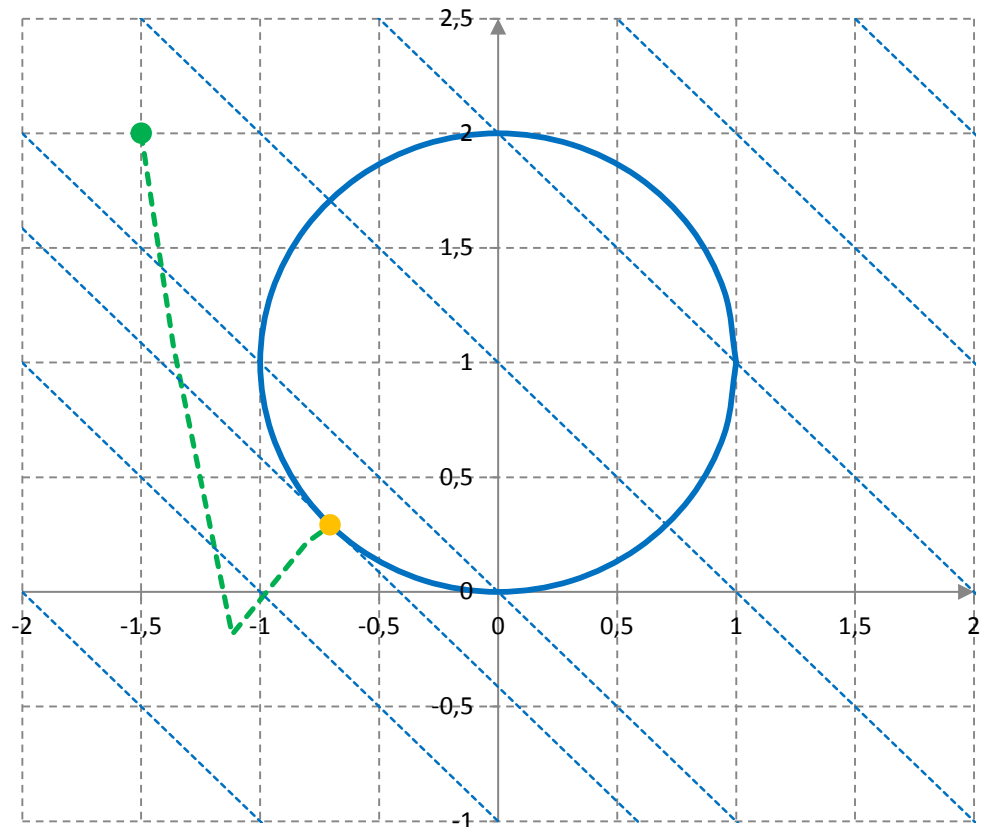
Exemple 2 : SQP sans globalisation

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} -1.5 \\ 2 \\ 1 \end{pmatrix}$

- Algorithme SQP sans globalisation
 → convergence rapide

Itération	x1	x2	λ
1	-1,50000	2,00000	1,00000
2	-1,36538	1,07692	0,42308
3	-1,11784	-0,18542	0,44290
4	-0,80352	0,21615	0,57183
5	-0,70990	0,28607	0,68889
6	-0,70718	0,29293	0,70697
7	-0,70711	0,29289	0,70711



3.5.4 Exemples

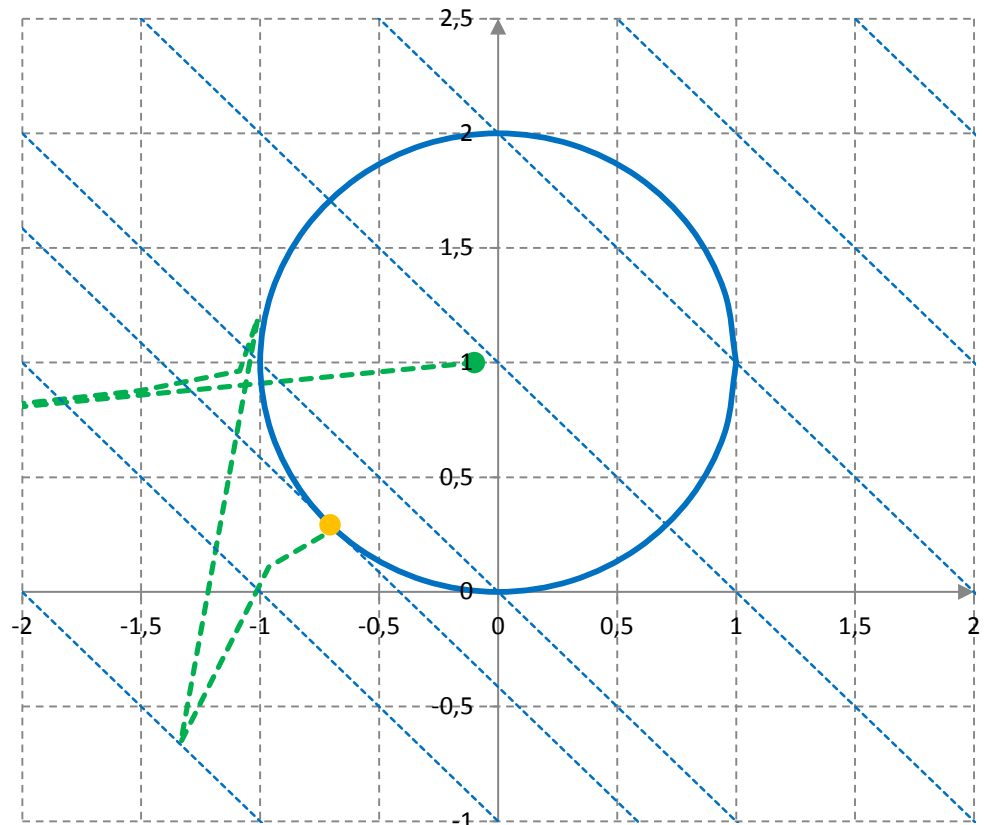
Exemple 2 : SQP sans globalisation

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} -0.1 \\ 1 \\ 1 \end{pmatrix}$

- Algorithme SQP sans globalisation
 → **convergence lente**

Itération	x1	x2	λ
1	-0,10000	1,00000	1,00000
2	-5,05000	0,50000	-44,50000
3	-2,62404	0,75032	-21,27825
4	-1,50286	0,87826	-8,90106
5	-1,08612	0,96364	-2,13558
6	-1,01047	1,19247	0,31161
7	-1,33383	-0,65614	0,39510
8	-0,96379	0,10912	0,48447
9	-0,72273	0,25387	0,63996
10	-0,70890	0,29344	0,70407
11	-0,70710	0,29289	0,70710
12	-0,70711	0,29289	0,70711



3.5.4 Exemples

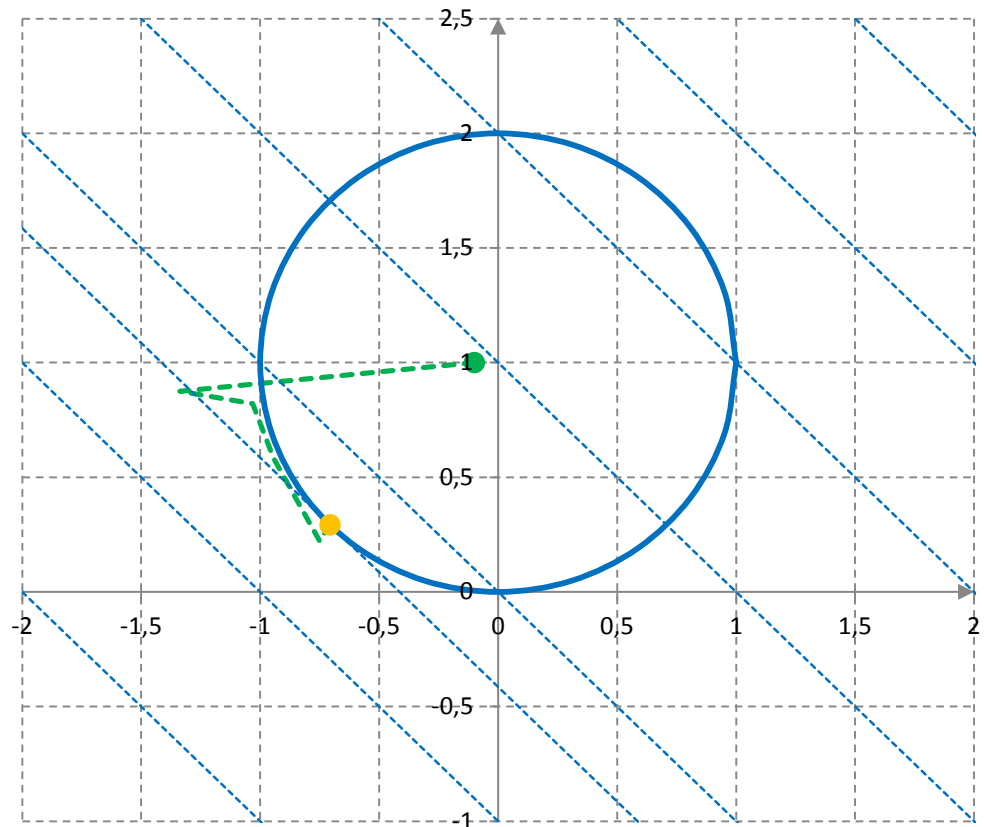
Exemple 2 : SQP avec globalisation

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} -0.1 \\ 1 \\ 1 \end{pmatrix}$

- Algorithme SQP avec globalisation
 → convergence plus rapide

Itération	x1	x2	λ
1	-0,10000	1,00000	1,00000
2	-1,33750	0,87500	-44,50000
3	-1,03171	0,82117	1,63129
4	-0,94371	0,58304	0,62377
5	-0,74975	0,22132	0,65803
6	-0,71035	0,29156	0,70147
7	-0,70710	0,29288	0,70708
8	-0,70711	0,29289	0,70711



3.5.4 Exemples

Exemple 2 : SQP avec globalisation

Itération	x_1	x_2	λ	∇L	$\nabla^2 L(x_1, x_2, \lambda)$		τ	dx	Pas s
1	-0,10000	1,00000	1,00000	9,90000	2,00000	0,00000	0,0	-4,95000	0,25
				1,00000	0,00000	2,00000	0,0	-0,50000	0,25
2	-1,33750	0,87500	-44,50000	-3,36370	-89,00000	0,00000	100,0	0,30579	1,00
				0,59218	0,00000	-89,00000	100,0	-0,05383	1,00
3	-1,03171	0,82117	1,63129	-0,28710	3,26258	0,00000	0,0	0,08800	1,00
				0,77690	0,00000	3,26258	0,0	-0,23812	1,00
4	-0,94371	0,58304	0,62377	-0,24198	1,24754	0,00000	0,0	0,19396	1,00
				0,45126	0,00000	1,24754	0,0	-0,36172	1,00
5	-0,74975	0,22132	0,65803	-0,05185	1,31605	0,00000	0,0	0,03940	1,00
				-0,09244	0,00000	1,31605	0,0	0,07024	1,00
6	-0,71035	0,29156	0,70147	-4,6E-03	1,40294	0,00000	0,0	3,2E-03	1,00
				-1,9E-03	0,00000	1,40294	0,0	1,3E-03	1,00
7	-0,70710	0,29288	0,70708	4,7E-06	1,41417	0,00000	0,0	-3,3E-06	1,00
				-1,7E-05	0,00000	1,41417	0,0	1,2E-05	1,00
8	-0,70711	0,29289	0,70711	-4,2E-10	1,41421	0,00000	0,0	3,0E-10	1,00
				2,7E-10	0,00000	1,41421	0,0	-1,9E-10	1,00

3.5.4 Exemples

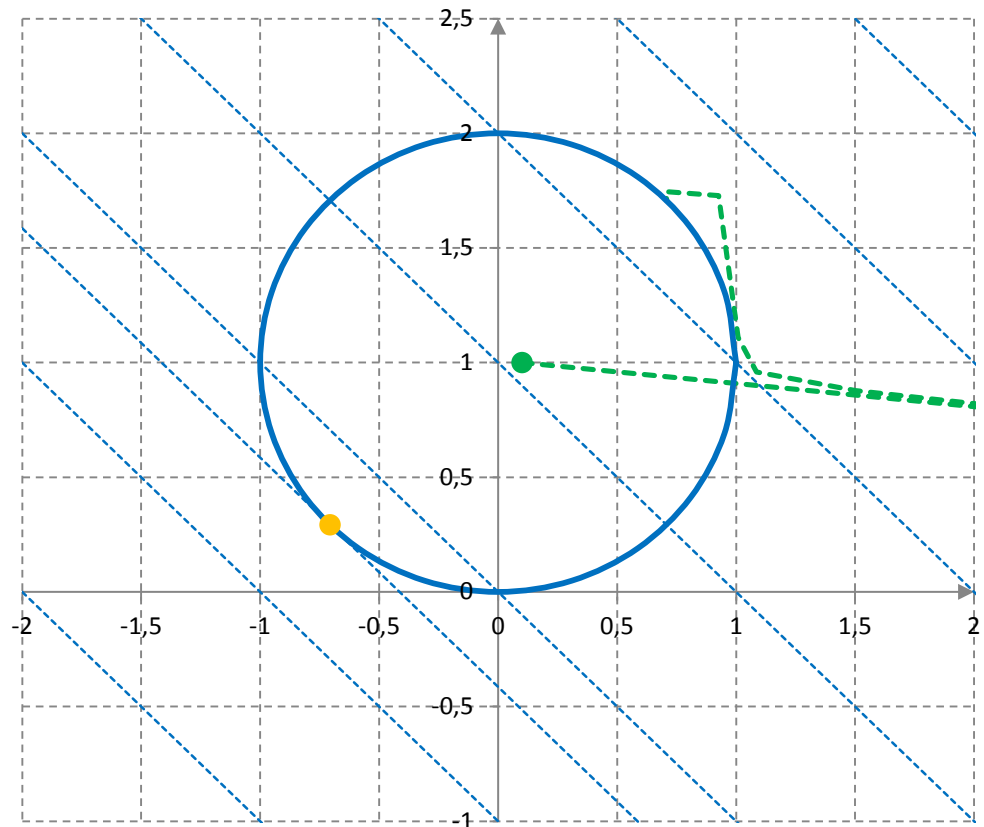
Exemple 2 : SQP sans globalisation

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 0.1 \\ 1 \\ 1 \end{pmatrix}$

- Algorithme SQP sans globalisation
 → **solution = maximum local**

Itération	x1	x2	λ
1	0,10000	1,00000	1,00000
2	5,05000	0,50000	-54,50000
3	2,62404	0,75028	-26,28019
4	1,50282	0,87782	-11,41974
5	1,08576	0,95907	-3,50192
6	1,00831	1,11015	-0,71030
7	0,92650	1,72824	-0,55351
8	0,70291	1,74580	-0,67324
9	0,70870	1,70662	-0,70579
10	0,70711	1,70711	-0,70710



3.5.4 Exemples

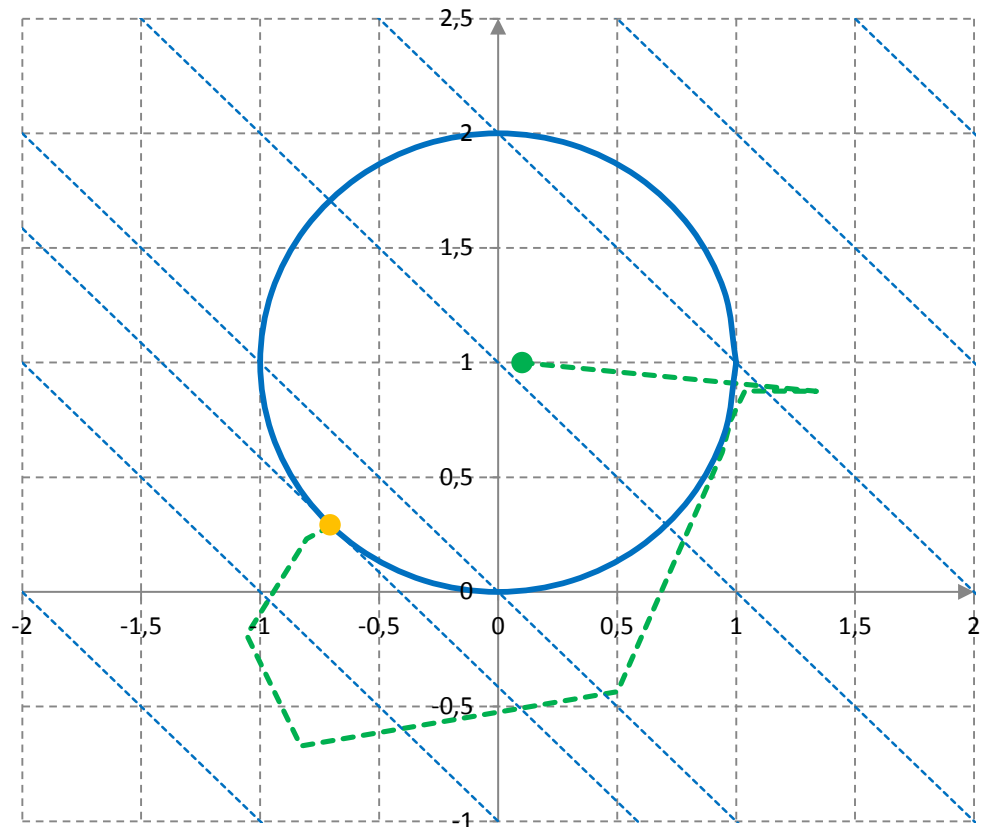
Exemple 2 : SQP avec globalisation

- Minimisation de $f(x) = x_1 + x_2$ sous $c(x) = x_1^2 + (x_2 - 1)^2 - 1 = 0$

- Point initial : $\begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 0.1 \\ 1 \\ 1 \end{pmatrix}$

- Algorithme SQP **avec globalisation**
 → **convergence**

Itération	x1	x2	λ
1	0,10000	1,00000	1,00000
2	1,33750	0,87500	-54,50000
3	1,03687	0,87643	4,23389
4	0,97837	0,75123	-0,24333
5	0,94133	0,60794	-0,35556
6	0,50173	-0,43482	-0,26135
7	-0,82925	-0,67191	0,26961
8	-1,05655	-0,18790	0,45516
9	-0,80511	0,23137	0,58156
10	-0,70800	0,28512	0,69118
11	-0,70721	0,29295	0,70699
12	-0,70711	0,29289	0,70711



3.5.4 Exemples

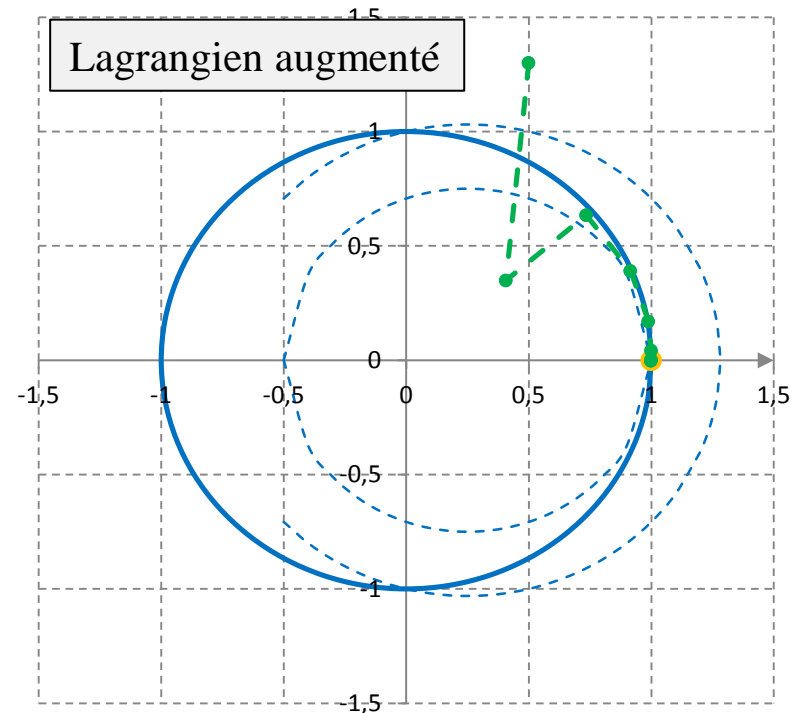
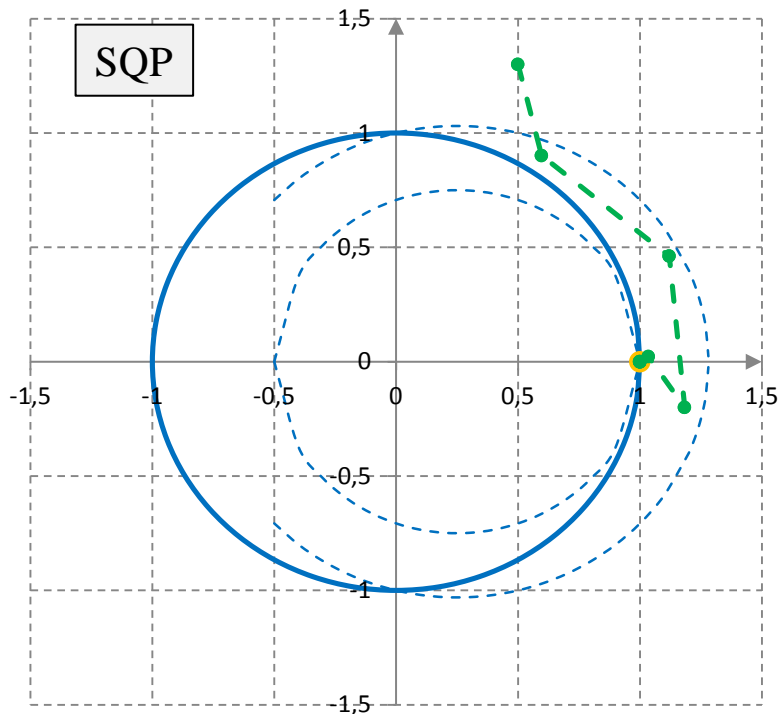
Exemple 2 : SQP avec globalisation

Itération	x_1	x_2	λ	∇L	$\nabla^2 L(x_1, x_2, \lambda)$		τ	dx	Pas s
1	0,10000	1,00000	1,00000	-9,90000	2,00000	0,00000	0,0	4,95000	0,25
				1,00000	0,00000	2,00000	0,0	-0,50000	0,25
2	1,33750	0,87500	-54,50000	12,32567	-109,00000	0,00000	150,0	-0,30063	1,00
				-0,05847	0,00000	-109,00000	150,0	0,00143	1,00
3	1,03687	0,87643	4,23389	0,49539	8,46779	0,00000	0,0	-0,05850	1,00
				1,06014	0,00000	8,46779	0,0	-0,12520	1,00
4	0,97837	0,75123	-0,24333	0,30426	-0,48667	0,00000	1,0	-0,59272	0,06
				1,17691	0,00000	-0,48667	1,0	-2,29267	0,06
5	0,94133	0,60794	-0,35556	0,50796	-0,71112	0,00000	1,0	-1,75838	0,25
				1,20493	0,00000	-0,71112	1,0	-4,17104	0,25
6	0,50173	-0,43482	-0,26135	1,27054	-0,52271	0,00000	1,0	-2,66197	0,50
				0,22632	0,00000	-0,52271	1,0	-0,47418	0,50
7	-0,82925	-0,67191	0,26961	0,24512	0,53921	0,00000	0,0	-0,45458	0,50
				-0,52197	0,00000	0,53921	0,0	0,96802	0,50
8	-1,05655	-0,18790	0,45516	-0,22888	0,91032	0,00000	0,0	0,25143	1,00
				-0,38167	0,00000	0,91032	0,0	0,41927	1,00
9	-0,80511	0,23137	0,58156	-0,11295	1,16311	0,00000	0,0	0,09711	1,00
				-0,06252	0,00000	1,16311	0,0	0,05376	1,00
10	-0,70800	0,28512	0,69118	-1,1E-03	1,38235	0,00000	0,0	8,0E-04	1,00
				-1,1E-02	0,00000	1,38235	0,0	7,8E-03	1,00
11	-0,70721	0,29295	0,70699	-1,4E-04	1,41398	0,00000	0,0	1,0E-04	1,00
				8,0E-05	0,00000	1,41398	0,0	-5,7E-05	1,00
12	-0,70711	0,29289	0,70711	1,2E-08	1,41421	0,00000	0,0	-8,4E-09	1,00
				-2,5E-08	0,00000	1,41421	0,0	1,8E-08	1,00

3.5.4 Exemples

Exemple 3 : Comparaison SQP – Lagrangien augmenté

- Minimisation de $f(x) = 2(x_1^2 + x_2^2 - 1) - x_1$ sous $c(x) = x_1^2 + x_2^2 - 1 = 0$
- Point initial : $x = \begin{pmatrix} 0.5 \\ 1.3 \end{pmatrix}$, $\lambda = 0$ Solution : $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\lambda = -\frac{3}{2}$



3.5.4 Exemples

Exemple 3 : Comparaison SQP – Lagrangien augmenté

Itération	SQP			Lagrangien augmenté		
	x_1	x_2	λ	x_1	x_2	λ
1	0,50000	1,30000	0,00000	0,50000	1,30000	0,00000
2	0,59665	0,90129	-1,38660	0,40707	0,34917	-0,71238
3	1,12042	0,46118	-1,70047	0,73467	0,63433	-1,29122
4	1,18366	-0,19988	-1,57065	0,91556	0,39077	-1,38175
5	1,03482	0,02190	-1,52359	0,98869	0,16985	-1,38175
6	1,00084	-0,00103	-1,50118	0,99953	0,04158	-1,30283
7	1,00000	0,00000	-1,50000	0,99905	-0,00320	-1,49103
8				0,99995	0,00171	-1,50003
9				1,00000	0,00045	-1,50003

Convergence rapide, très précise

Convergence plus lente, moins précise
 → suivi de la contrainte
 → conditionnement

Sommaire

1. Bases théoriques
2. Optimisation sans contraintes
- 3. Optimisation avec contraintes**
 - 3.1 Simplexe
 - 3.2 Point intérieur
 - 3.3 Gradient projeté
 - 3.4 Lagrangien augmenté
 - 3.5 Programmation quadratique séquentielle
 - 3.6 Convergence**
 - 3.6.1 Convergence globale
 - 3.6.2 Comparaison des algorithmes
 - 3.6.3 Difficultés pratiques

3.6.1 Convergence globale

Problème sans contraintes

$$\min_{x \in \mathbb{R}^n} f(x)$$

Convergence globale

- Un algorithme générant une suite $(x_k)_{k \in \mathbb{N}}$ à partir du point x_0 est **globalement convergent** si

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$$

- Convergence globale = convergence vers un point stationnaire
≠ convergence vers le minimum global
- Le **domaine de convergence** est l'ensemble des points initiaux x_0 à partir desquels on a la convergence globale.
- Le domaine de convergence peut être très réduit lorsque :
 - la fonction est très non linéaire
 - la fonction est mal conditionnée
 - difficulté principale de la méthode de Newton

3.6.1 Convergence globale

Descente avec recherche linéaire

- Un algorithme à base de **recherche linéaire** génère une suite $(x_k)_{k \in \mathbb{N}}$ à partir du point x_0 :
 - suivant des directions $d_k \rightarrow x_{k+1} = x_k + s_k d_k$
 - avec des pas s_k
- On suppose que :
 - les directions d_k sont des **directions de descente** $\rightarrow \nabla f(x_k)^T d_k < 0$
 - les pas s_k vérifient les **conditions de Wolfe** $\rightarrow \begin{cases} f(x_k + s d_k) < f(x_k) + c_1 s \nabla f(x_k)^T d_k \\ \nabla f(x_k + s d_k)^T d_k > c_2 \nabla f(x_k)^T d_k \end{cases}$
 - le gradient de la fonction est continu au sens de Lipschitz

- Si les directions ne deviennent pas « trop » orthogonales au gradient : $-\frac{\nabla f(x_k)^T d_k}{\|\nabla f(x_k)\| \|d_k\|} \geq \varepsilon, \forall k$
 alors l'algorithme est globalement convergent pour tout $x_0 \in \mathbb{R}^n$: $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$

- condition vérifiée par :
- la méthode de **plus forte pente**
 - la méthode de **Newton** si le conditionnement du hessien est borné
 - les méthodes incluant des **itérations périodiques de plus forte pente**
- méthodes globalement convergentes

3.6.1 Convergence globale

Eléments de la démonstration

θ_k = angle entre le gradient $\nabla f(x_k)$ et la direction de descente d_k

- **Théorème de Zoutendijk** : $\sum_k \cos^2 \theta_k \|\nabla f_k\|^2 < \infty$ avec $\cos \theta_k = -\frac{\nabla f_k^T d_k}{\|\nabla f_k\| \|d_k\|}$

Démontré à partir de : $\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$, $\forall x, y$ (condition de Lipschitz)

$$\text{et } \begin{cases} f(x_k + sd_k) < f(x_k) + c_1 s \nabla f(x_k)^T d_k \\ \nabla f(x_k + sd_k)^T d_k > c_2 \nabla f(x_k)^T d_k \end{cases} \quad (\text{conditions de Wolfe})$$

$$\Rightarrow \text{ou } \begin{cases} \cos^2 \theta_k \rightarrow 0 \\ \|\nabla f_k\| \rightarrow 0 \end{cases}$$

- Méthode de **plus forte pente** : $d_k = -\nabla f_k \Rightarrow \cos \theta_k = 1 \Rightarrow \lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$
- Méthode de **Newton** : $d_k = -H_k^{-1} \nabla f_k$ avec $\|H_k\| \|H_k^{-1}\| < M$ (conditionnement borné)

En utilisant le théorème de Rayleigh-Ritz : $\sigma_1 = \max_{x \neq 0} \frac{x^T A x}{x^T x}$, $\sigma_n = \min_{x \neq 0} \frac{x^T A x}{x^T x}$
 (valeurs propres $\sigma_1 \geq \dots \geq \sigma_n$)

$$\Rightarrow \cos \theta_k > 1/M \Rightarrow \lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$$

3.6.2 Comparaison des algorithmes

Classification

- Méthodes primales
- Méthodes primales-duales
- Méthodes duales

	Méthode primale	Méthode primale-duale	Méthode duale
Problème traité	problème primal	problème primal	problème dual
Objectif	min f - méthode directe - point stationnaire	solution KKT - méthode indirecte - point stationnaire	max w - méthode indirecte - point col
Itérations	admissibles	admissibles ou non	non admissibles
Variables	primales x	primales x , duales λ	primales x , duales λ
Algorithmes	- simplexe (LP) - gradient projeté - pénalisation	- point intérieur (LP, NLP) - séquentiel quadratique - lagrangien augmenté	- Uzawa

3.6.2 Comparaison des algorithmes

Méthode (auteurs, date)	Type	Convergence	Avantages	Difficultés
Plus forte pente (Cauchy 1847)	sans contraintes	linéaire	convergence globale	convergence lente si mauvais conditionnement
Quasi-Newton (DFP 1959, BFGS 1970)	sans contraintes	superlinéaire quadratique	1 minimisation par itération	hessien défini positif stockage mémoire (n^2)
Gradient conjugué (Hestenes 1951, Stiefel 1952) (Fletcher-Reeves 1964)	sans contraintes	superlinéaire quadratique	stockage mémoire réduit	hessien défini positif n minimisations par itération
Simplexe (Dantzig 1950)	problème linéaire	exponentiel	vitesse de convergence itérations admissibles	algorithme exponentiel dégénérescence
Gradient projeté, réduit (Rosen 1960, Wolfe 1962) (Abadie-Carpentier 1969)	primale		itérations admissibles	convergence lente si - mauvais conditionnement - non linéarités fortes
Séquentiel quadratique	primale-duale	quadratique (Newton)	vitesse de convergence	globalisation nécessaire itérations non admissibles
Pénalisation extérieure (Fiacco-Mc Cormick 1968)	primale	quadratique (Newton)	sans contraintes	précision contraintes conditionnement
Pénalisation intérieure Point intérieur (Karmarkar 1984)	primale-duale problème linéaire	quadratique (Newton)	algorithme polynomial (LP)	précision contraintes
Lagrangien augmenté (Hestenes 1969, Powell 1969)	primale-duale	quadratique (Newton)	sans contraintes	précision contraintes
Uzawa (1958)	duale	linéaire	sans contraintes	précision contraintes

3.6.3 Difficultés pratiques

Qualité du minimum recherché

- Minimum global approché → méthodes aléatoires (heuristiques), dites « globales »
- **Minimum local précis** → méthodes déterministes (gradient), dites « locales »

Principales difficultés des méthodes locales

- **Initialisation** : le minimum est recherché au voisinage du point initial.
 - Essayer plusieurs points initiaux
 - Evaluer la qualité de la solution obtenue (connaissance expérimentale du problème)
- **Contraintes actives** : il faut identifier à chaque itération les contraintes actives (→ égalité)
 - Stratégie d'activation / désactivation des contraintes en fonction du déplacement
- **Précision numérique** : l'évaluation du critère et des contraintes est bruitée.
 - Adapter les incréments pour les gradients
 - Adapter les seuils de précision et les tests d'arrêts
- **Conditionnement** : les valeurs numériques doivent être du même ordre de grandeur
 - Mise à l'échelle des variables et des fonctions (critère, contraintes)
 - Mise à l'échelle des dérivées premières (jacobien) et secondes (hessien)
 - On ne peut pas tout mettre à l'échelle simultanément → essais

Résumé

Bases théoriques

- Différentiabilité : gradient, hessien, jacobien, lignes de niveau (§1.1.3)
- Contraintes linéaires : base, solution de base, direction de base (§1.2.2)
- Conditions d'optimalité : lagrangien, conditions KKT (§1.4.3)

Optimisation sans contraintes

- Méthode de Newton : principe, quasi-Newton, globalisation (§2.2)
- Moindres carrés linéaires : équations normales (§2.5.3)

Optimisation avec contraintes

- Programmation linéaire : simplexe (§3.1.3)
 - méthode des tableaux, méthode des 2 phases
- Programmation non linéaire : gradient projeté/réduit (§3.3)
 - lagrangien augmenté (§3.4)
 - SQP (§3.5)
 - principes de chaque algorithme

Difficultés pratiques

- Minima locaux
- Précision numérique

Bibliographie

Livres en français

- **Programmation mathématique** (M. Minoux)
 - **Introduction à l'optimisation différentiable** (M. Bierlaire)
 - Optimisation continue (F. Bonnans)
 - Optimisation discrète (A. Billionnet)
-

Livres en anglais

- **Practical optimization** (P. E. Gill, W. Murray, M. H. Wright)
- **Numerical optimization** (J. Nocedal, S. J. Wright)
- **Practical methods of optimization** (R. Fletcher)
- Practical mathematical optimization (J. A. Snyman)
- Numerical optimization (J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, C. A. Sagastizabal)
- Numerical recipes in Fortran (W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery)
- Numerical methods for unconstrained optimization and nonlinear equations (J. E. Dennis, R. B. Schnabel)