Introduction à l'optimisation

MU4MEN01 : CALCUL SCIENTIFIQUE, OPTIMISATION – 3 ECTS

RESPONSABLE D'UE ET COURS : FLORENCE OSSART

Cours du 15/09/2021

- → Présentation de l'UE
- → Explication du déroulement des TD/TP sur machine
- → Ch. 1 : Introduction, généralités
- → Ch. 2 : Minimisation sans contrainte dans $\mathbb{R}^n$

Objectifs de l’enseignement

→ Concepts fondamentaux de l’optimisation numérique :

- Analyser et formuler mathématiquement un problème d’optimisation
- Connaître les différents types de problème d’optimisation
- Connaître *quelques* méthodes d’optimisation continue classiques

→ Programmation :

- Développer ses compétences en algorithmique
- Développer ses compétences en langage python

→ Projet d’application :

- Mettre en œuvre ses compétences théoriques et pratiques
- Vérifier et analyser les résultats obtenus

Organisation de l'UE

<table>
<tr><th>Semaine</th><th>Cours</th><th>TP</th><th>Projet</th></tr>
<tr><td>15 septembre</td><td>Cours 1</td><td></td><td></td></tr>
<tr><td>20 au 25 septembre</td><td></td><td>TP1</td><td></td></tr>
<tr><td>29 septembre</td><td>Cours 2</td><td></td><td></td></tr>
<tr><td>4 au 8 octobre</td><td></td><td>TP2</td><td></td></tr>
<tr><td>11 au 15 octobre</td><td></td><td>TP3</td><td rowspan="5">Etude complète d'un « vrai » problème d'optimisation.
Travail en équipe de 3 étudiants du même groupe de TP.
En autonomie, chargé de TP disponible pour répondre aux questions par mail</td></tr>
<tr><td>20 octobre</td><td>Cours 3</td><td></td></tr>
<tr><td>25 au 29 octobre</td><td></td><td>TP4</td></tr>
<tr><td>10 novembre</td><td>Cours 4</td><td></td></tr>
<tr><td>Jusqu'au 10 décembre</td><td></td><td></td></tr>
</table>

Documents et communication

→ Sur moodle :

- Slides de cours et autres documents
- Si vous n'êtes pas inscrit : m'envoyer un mail avec nom, prénom, n° d'étudiant (préciser qu'il s'agit du cours d'optim)

→ Par mail :

- florence.ossart@sorbonne-universite.fr

Fonctionnement des séances de TP

→ *Travail en salle machine*. Utilisation des PC de la salle ou de son PC perso.

→ *Programmation en python* (voir cours de programmation orientée objets)

→ *Utilisation de notebook*, outil adapté à la rédaction de comptes rendus de travaux numériques

→ *Outils utilisés et à installer impérativement sur votre PC personnel* : python 3 et jupyter pour l'édition de notebook

Principe du notebook

→ Document constitué d'un assemblage de cellules (cells) de texte et de code informatique.

→ Les cellules de texte, éditées en langage 'markdown' (basé sur des balises), sont utilisées pour décrire l'énoncé du problème, expliquer la démarche de résolution et commenter les résultats obtenus.

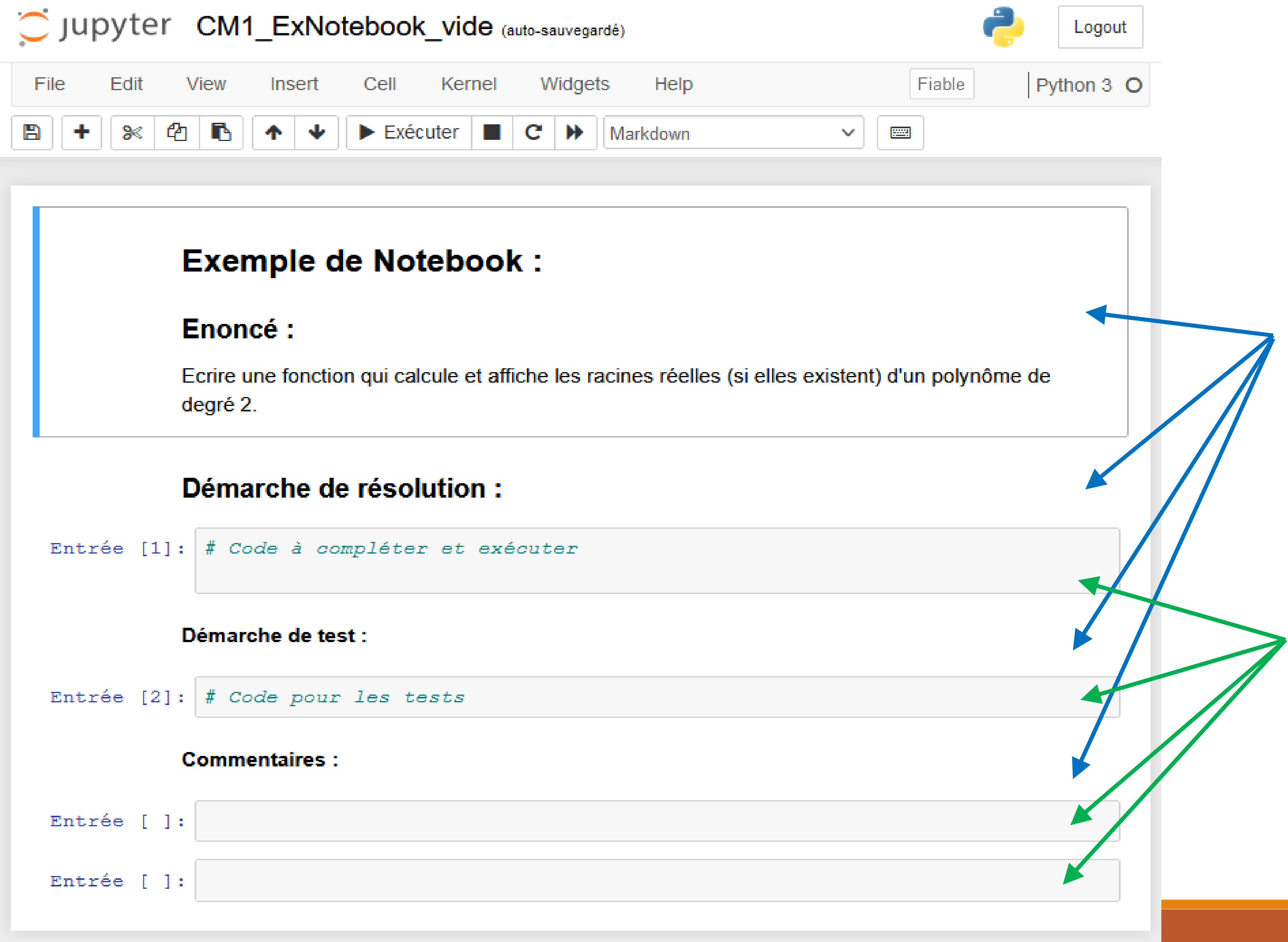
→ Les cellules de code contiennent du code informatique qui peut être exécuté. Elles sont utilisées pour traiter numériquement un certain problème.

→Grâce à cet outil, il est facile de rédiger un compte rendu complet de la démarche de résolution mise en œuvre et des différents calculs réalisés afin de répondre au problème posé.

→ Deux liens parmi beaucoup d'autres pour en savoir plus :

- https://openclassrooms.com/fr/courses/4452741-decouvrez-les-librairies-python-pour-la-data-science/5574801-faites-vos-premiers-pas-dans-un-notebook-jupyter
- https://python.sdv.univ-paris-diderot.fr/18_jupyter/

Sujet brut, à compléter



Cellules de texte

Cellules de code

jupyter CM1_ExNotebook Dernière Sauvegarde : 19/07/2021 (auto-sauvegardé)

Logout

File Edit View Insert Cell Kernel Widgets Help

Fiable | Python 3

Exécuter | Code

Sujet complété

Exemple de Notebook :

Enoncé :

Ecrire une fonction qui calcule et affiche les racines réelles (si elles existent) d'un polynôme de degré 2.

Cellules de texte

Solution :

Principe :

Soit P le polynôme de degré 2 défini par : $P(x) = ax^2 + bx + c$.

Pour calculer les racines de P, il faut calculer le discriminant Δ, défini par $\Delta = b^2 - 4ac$. Si Δ est strictement positif, il existe deux racines réelles distinctes : $x_1 = \frac{-b-\sqrt{\Delta}}{2a}$ et $x_2 = \frac{-b+\sqrt{\Delta}}{2a}$. Si Δ est nul, les deux racines sont confondues : $x_1 = x_2 = \frac{-b}{2a}$. Si Δ est strictement négatif, il n'existe pas de racine *réelle*.

Code de la fonction *racine_P2* :

Cellules de code

Entrée [1]:

```
import math

def racine_P2(a,b,c) :
    """ Calcul et affichage des racines d'un polynome de degré 2
    @param a coefficient du terme de degré 2
    @param b coefficient du terme de degré 1
    @param c coefficient du terme de degré 0
    @author Florence Ossart
    @version 1.0
    """
    # Entree : a, b, c coefficients du polynome
    # Sortie : aucune
    print('Racines réelles du polynome {}x^2 + {}x + {} : '.format(a,b,c))
    delta = b**2 - 4*a*c
    if delta > 0 :
```

Le langage Markdown

→ Mise en forme d'un texte grâce à des balises qui encadrent le texte à mettre en forme.

Texte brut (Markdown)	Texte affiché
`Lisez l'énoncé *avant* de venir en TP.`	Lisez l'énoncé *avant* de venir en TP.
`Lisez l'énoncé **avant** de venir en TP.`	Lisez l'énoncé **avant** de venir en TP.
`# Titre 1`	**Titre 1**
`## Titre 2`	**Titre 2**
`Soit $f$, définie par $f(x)=ax^2$.`	Soit f, définie par $f(x) = ax^2$.

→ Deux liens parmi beaucoup d'autres pour en savoir plus :

- https://openclassrooms.com/fr/courses/1304236-redigez-en-markdown
- https://learninglab.gitlabpages.inria.fr/mooc-rr/mooc-rr-ressources/module1/ressources/introduction_to_markdown_fr.html#lettres-grecques

Evaluation du module

→ Note finale :

- 20% TP : évaluation d'un notebook fait en devoir maison
- 50% projet : évaluation du rapport (notebook ou autre format) – aspect démarche, code et analyse des résultats
- 30% écrit : examen sur les aspects théoriques des méthodes vues en cours et en TD/TP + 1 exercice dérivé du projet

→ Deuxième chance :

- Attention : 70% de la note est figée (TP + projet). La 2ème chance porte sur 30% de la note d'UE.

Des questions sur le fonctionnement du module ?

I - Introduction

DÉFINIR UN PROBLÈME D'OPTIMISATION

VOCABULAIRE

1.1 Définition

→ Un problème d'optimisation consiste à déterminer le meilleur élément parmi un ensemble donné, au sens d'un critère qualitatif.

→ Cela conduit à minimiser ou maximiser une fonction-coût sur un certain ensemble.

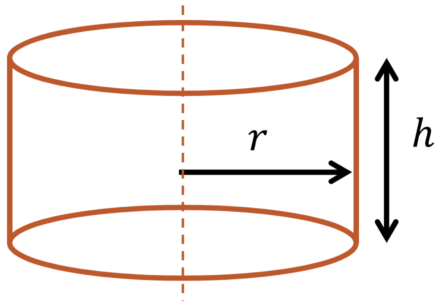
→ Domaines d'application :

- Chaîne logistique (production, distribution, …)
- Gestion des réseaux (électrique, téléphone, informatique, …)
- Planification de vols, trains (partage de ressources, satisfaction des usagers)
- Calcul de trajet (waze, google map, …)
- Contrôle optimal
- …

1.2 Commençons par un exemple simple

→ Situation :

- On veut produire des boîtes cylindriques de volume donné V_0 en utilisant le moins de matière possible.
- Il faut donc minimiser la surface du cylindre, à volume donné.



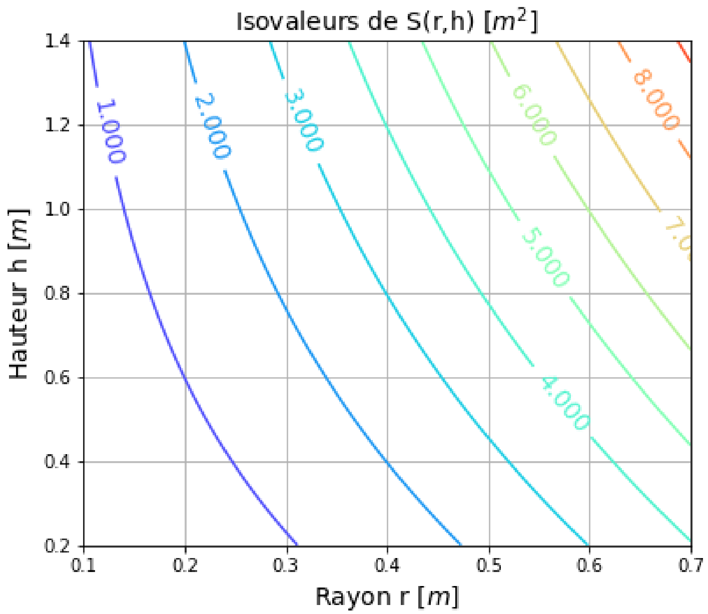
→ Formalisation mathématique :

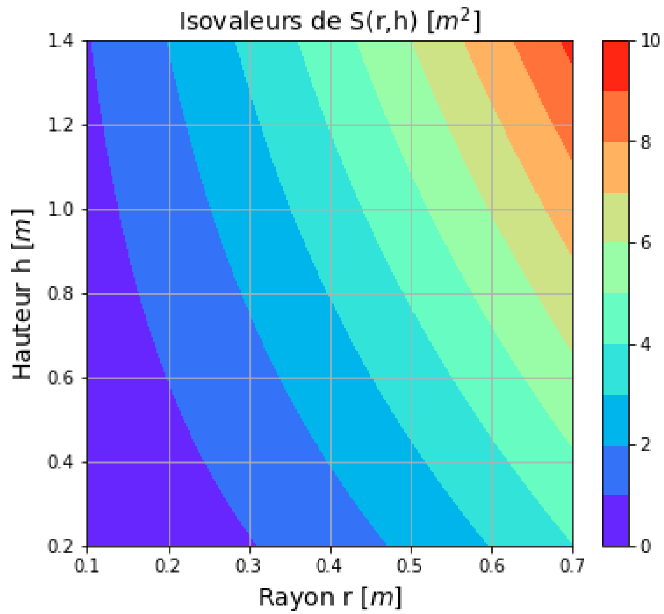
- Variables de décision : rayon r et hauteur h
- Fonction coût : surface $S(r,h) = 2\,\pi\,r^2 + 2\,\pi\,r\,h$
- Contrainte : volume imposé $\mathrm{V}(r,h) = \pi\,r^2\,h = V_0$

- Problème : déterminer (r^*, h^*) tel que : $\forall (r,h) \in \mathbb{R}^+ \times \mathbb{R}^+$ $\quad S(r^*,h^*) \leq S(r,h)$, $\quad \mathrm{V}(r,h) = V_0$

1.2.a Visualisation du critère « S(r,h) »

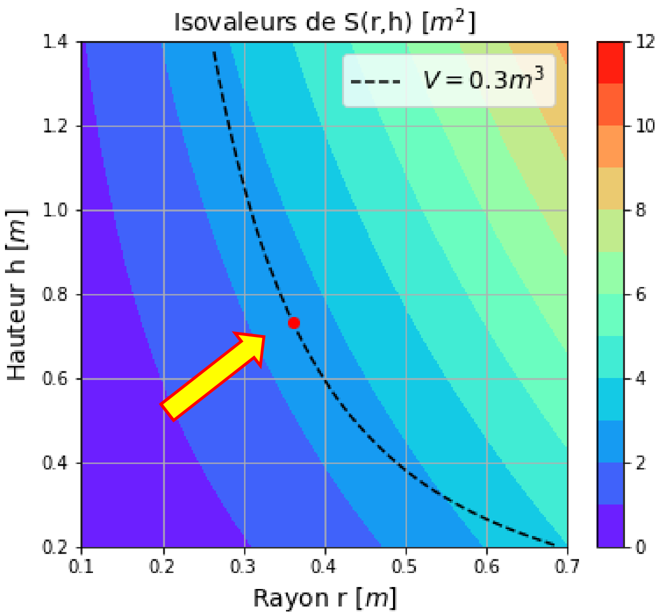
→ Bibliothèque matplotlib : fonctions « contour » (gauche) et « contourf » (droite)

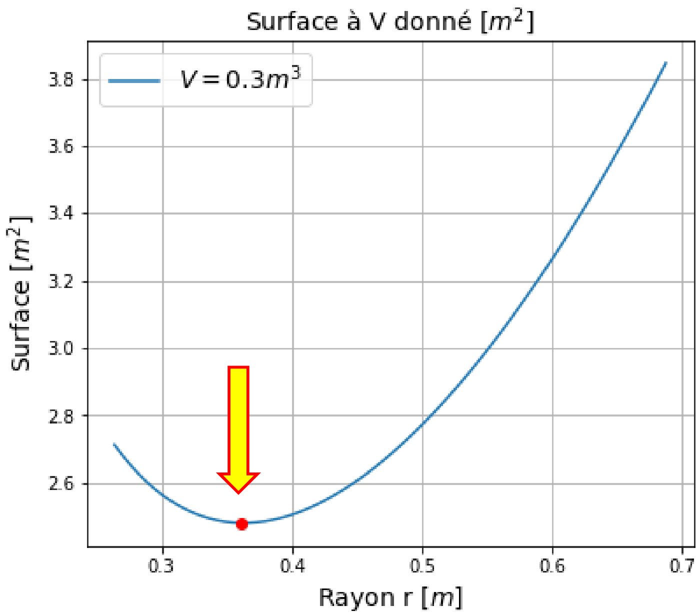




1.2.b Visualisation du problème contraint

→ La contrainte de volume transforme le critère S(r,h) en critère S(r,h(Vo,r))





1.2.c Résolution du problème contraint

→ La contrainte de volume transforme la fonction objectif à 2 variables $S(r, h)$ en une fonction objectif à 1 variable$S_{V_0}(r) = S\left(r, h(V_0)\right)$

→ Fonction étudiée : $S(r, h) = 2\,\pi\,r^2 + 2\,\pi\,r\,h,$

→ La contrainte $\pi\,r^2\,h = V_0$ implique que $h = \frac{V_0}{\pi\,r^2}$

→ La fonction à minimiser à V_0 donné est donc : $S_{V_0}(r) = 2\,\pi\,r^2 + \frac{2\,V_0}{r}$

1.2.c Résolution du problème contraint, 2

→ Fonction à minimiser est : $S_{V_0}(r) = 2\,\pi\, r^2 + \frac{2\,V_0}{r}$, pour $r > 0$

→ La dérivée première $\frac{d}{dr} S_{V_0}(r) = 4\,\pi\, r - \frac{2\,V_0}{r^2}$ s'annule pour $r^* = \sqrt[3]{\frac{V_0}{2\pi}}$

→ La dérivée seconde $\frac{d^2}{dr^2} S_{V_0}(r) = 4\,\pi + \frac{4\,V_0}{r^3}$ est positive $\forall r > 0$, donc r^* est un minimum

→ Tous calculs faits : $r^* = \sqrt[3]{\frac{V_0}{2\pi}}$ $\quad h = 2r^*$ $\quad S^* = 3\sqrt[3]{2\pi\, V_0^2}$

→ Pour $V_0 = 0{,}3\ m^2$, on obtient : $r^* = 0{,}36\ m$, $h = 0{,}72\ m$ et $S^* = 2{,}48\ m^2$

1.3 Formulation générale d'un problème d'optimisation

Trouver x^* tel que :

$J(x^*) = \inf \{ J(x), x \in A \}$

avec : $C_E(x) = 0$

$C_I(x) \leq 0$

→ Critère d'évaluation :

- Performance d'un jeu de variables de décision

$$J : \begin{array}{l} A \longrightarrow \mathbb{R} \\ x \longmapsto J(x) \end{array}$$

→ Variables de décision :

- Variables par rapport auxquelles se fait l'optimisation
- Notation : $x = (x_1, x_2, \dots, , x_n)$ $x \in A$
- A est le domaine admissible

→ Contraintes :

- Contraintes-égalités : $C_E(x) = 0$
- Contraintes-inégalités : $C_I(x) \leq 0$

1.3.a Les questions qu'on se pose :

Trouver x^* tel que :
$J(x^*) = \inf \{ J(x) , x \in A \}$
avec : $C_E(x) = 0$
$C_I(x) \leq 0$

→ Existe-t-il une solution ?

→ La solution est-elle unique ?

→ Comment trouver la solution ?

1.3.b Différents types de problèmes d'optimisation

→ Classement selon les caractéristiques des variables, de la fonction-coût et des contraintes

→ Variables : Continues / discrètes

→ Fonction-coût :

- Linéaire / Quadratique / Convexe / Coercive
- Continue / Différentiable (dérivées calculables?)
- Rapide / longue à calculer

→ Contraintes : Linéaires / convexes

→ A chaque type de problème correspondent des méthodes de résolution adaptées

1.3.c Types de fonctions

→ Linéaire (affine, en réalité)

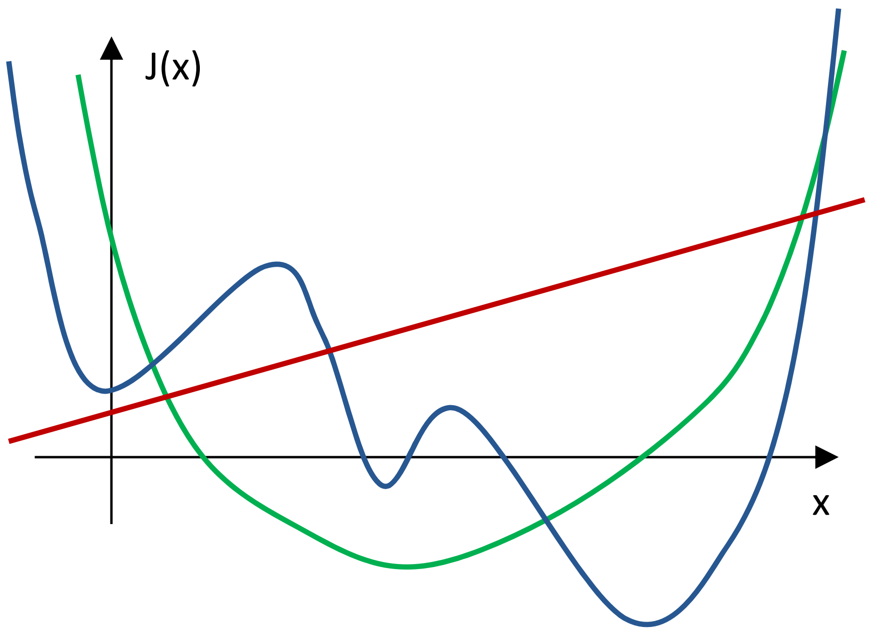
→ Quadratique (polynôme degré 2)

→ Convexe

- Dérivée seconde toujours positive

→ Coercive

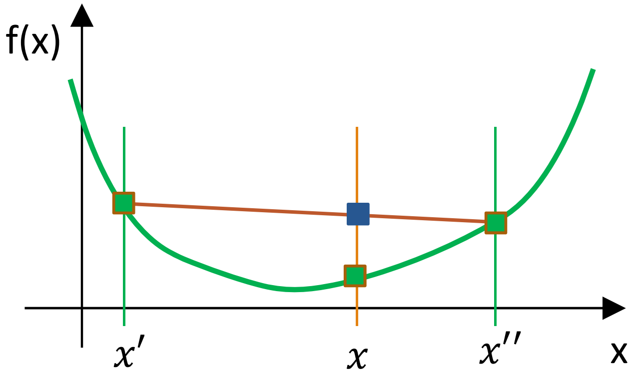
- Tend vers l'infini à l'infini
- $\lim_{\|x\| \to \infty} J(x) = \infty$



Schématisation 1D de différents types de fonctions

1.3.d Convexité

→ Fonction convexe :

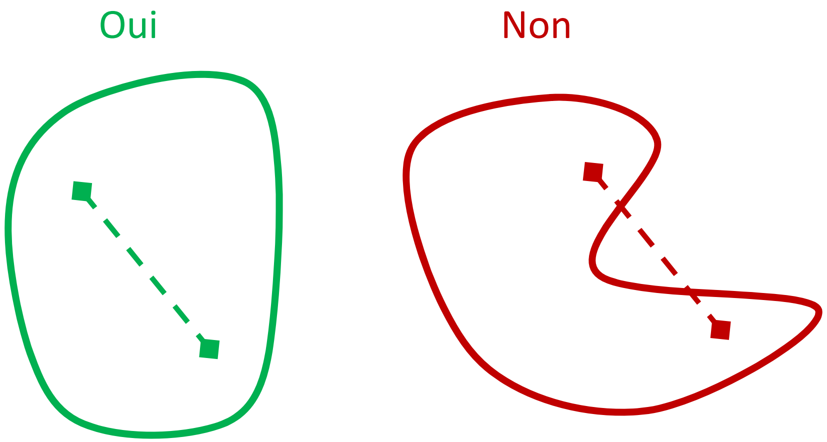


$$x' \leq x \leq x'' \Longrightarrow$$

$$f(x) \leq f(x') + \frac{f(x'') - f(x')}{x'' - x'}(x - x')$$

La corde est toujours au-dessus de la courbe

→ Domaine convexe : tout segment entre deux points intérieurs est entièrement contenu dans le domaine.



II. Minimisation sans contrainte dans $\mathbb{R}^n$

RAPPELS SUR LES FONCTIONS DE PLUSIEURS VARIABLES

II.1 Formulation du problème

Trouver x^* tel que :

$$J(x^*) = \inf \{ J(x), x \in A \}$$
$$A \subset \mathbb{R}^n$$

→ Critère d'évaluation :

- Performance d'un jeu de variables de décision

$$J: \begin{array}{l} A \longrightarrow \mathbb{R} \\ x \longmapsto J(x) \end{array}$$

→ Variables de décision :

- Variables par rapport auxquelles se fait l'optimisation
- Notation : $x = (x_1, x_2, \dots, , x_n)$ $x \in A$
- A est le domaine de définition de J

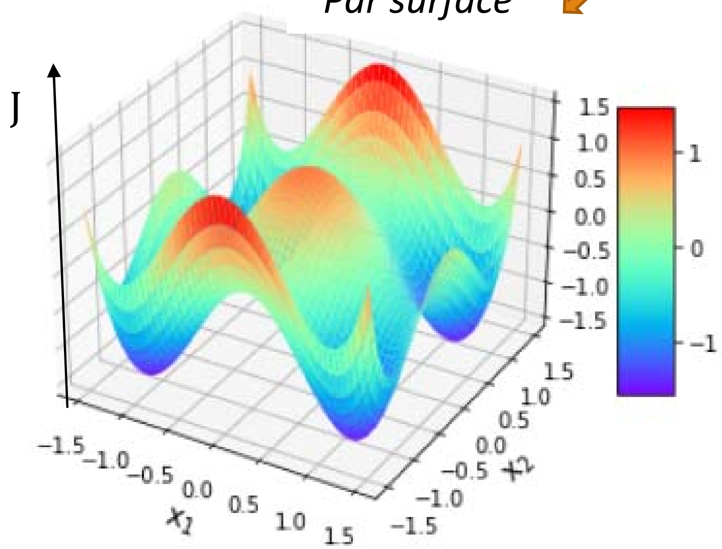
→ Il s'agit donc d'une « bête » recherche de minimum…

II.1.a Comment visualiser la fonction ?

→ On sait faire pour une fonction de deux variables : $\mathrm{J}(x_1, x_2)$, mais pas pour 3 variables et plus

Représentations complémentaires

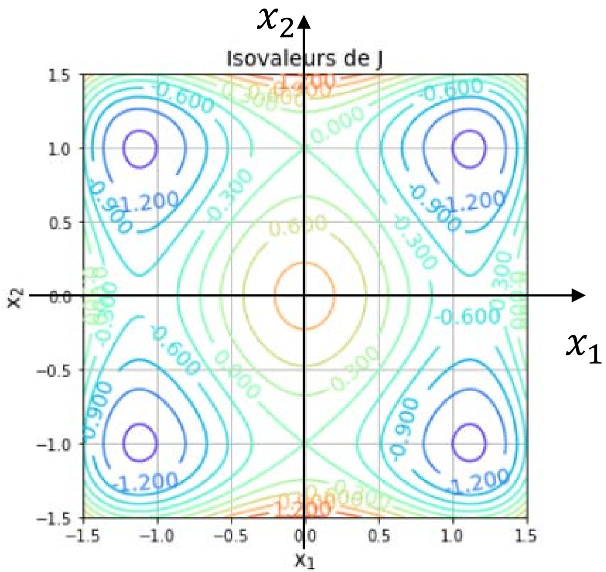
Par surface



Par isovaleurs :

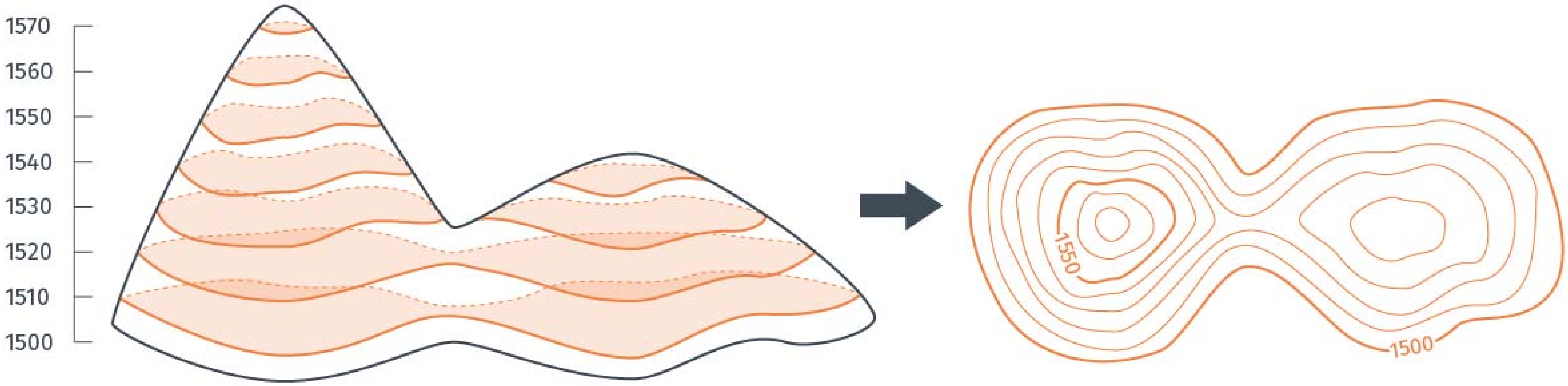
réseau de courbes

$\mathrm{J}(x_1, x_2) = k.\Delta J, k \in \mathbb{Z}$



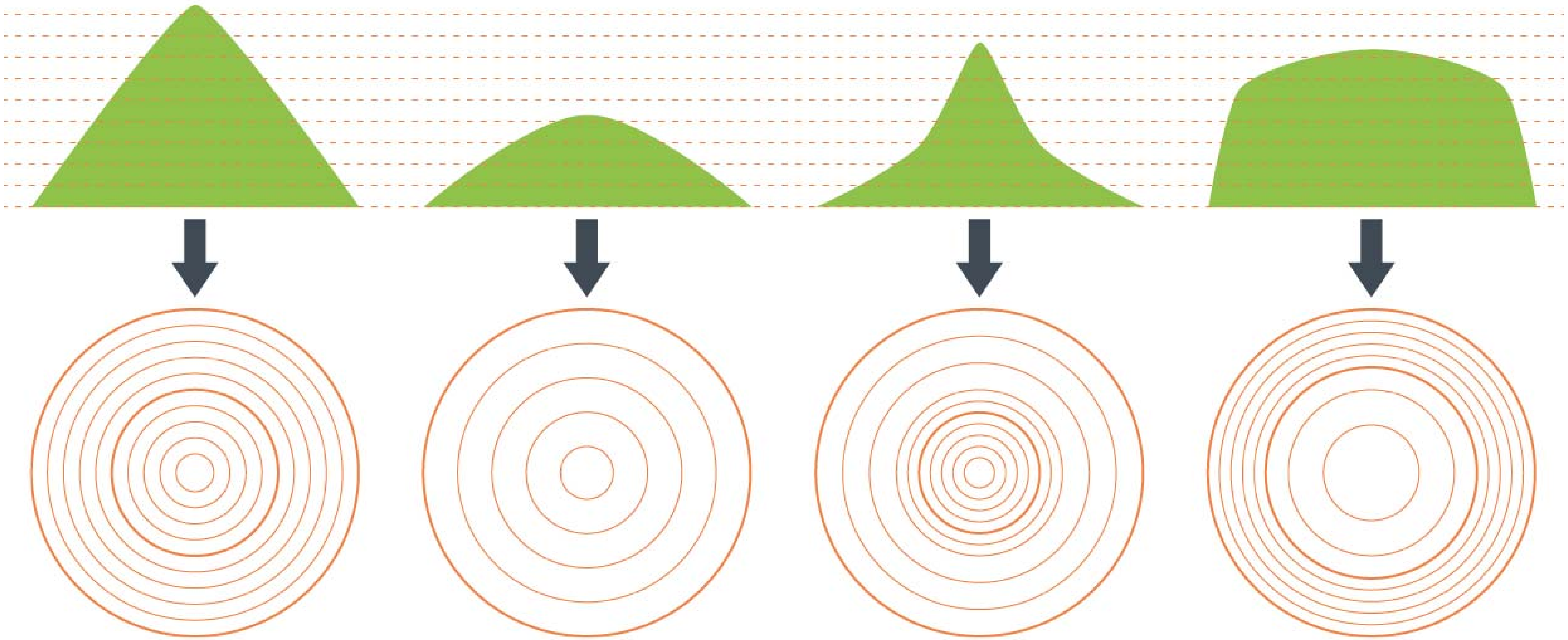
Exemple d’isovaleurs : les lignes de niveau

→ Utilisation en cartographie : projection d’informations 3D dans le plan



[figure extraite de https://www.ign.fr/reperes/apprendre-lire-une-carte-en-cinq-etapes

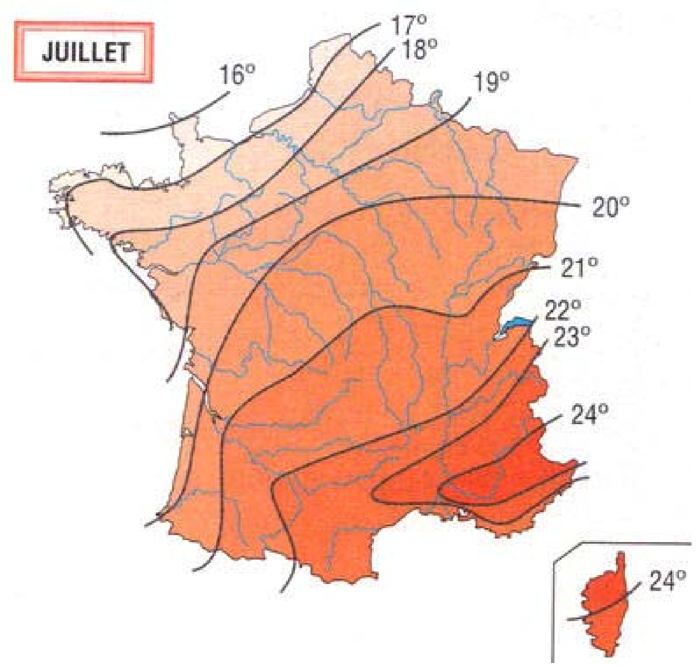
La distance entre ligne renseigne sur le gradient de la fonction représentée.



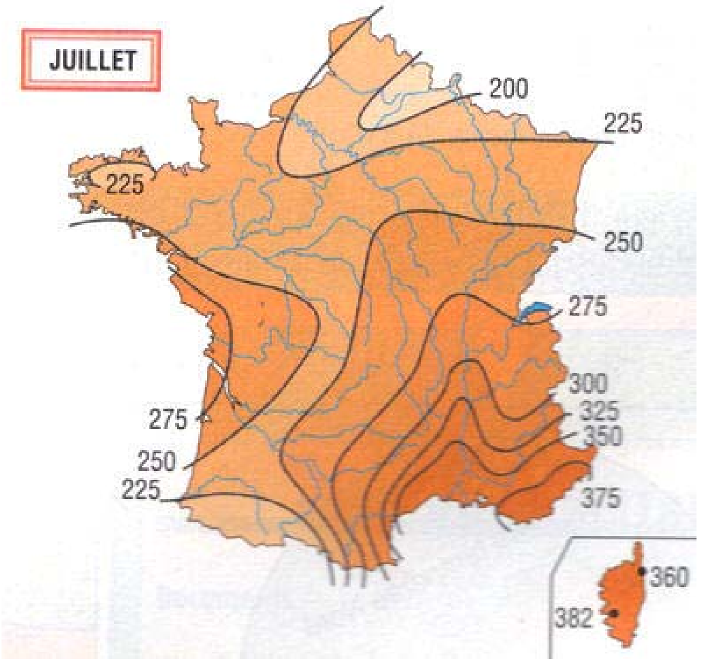
[figure extraite de https://www.ign.fr/reperes/apprendre-lire-une-carte-en-cinq-etapes

Autres exemples d'isovaleurs

Isothermes



Iso-ensoleillement (heures par mois)



[cartes extraites de http://www.alertes-meteo.com/cartes/carte-isotherme-juillet-france.php]

II.1.b Les questions qui se posent

→ Existence — Le problème admet-il une solution ?

→ Unicité — Si le problème admet une solution, est-elle unique ?

→ Conditions — Quelles équations faut-il résoudre pour trouver la solution ?

→ Algorithmes — Comment trouver numériquement la solution ?

II.2 Approche analytique

→ Hypothèses :

- La fonction $J(x)$ est continue et deux fois dérivable
- *On a accès à J et à ses dérivées*

Dérivée d'ordre 1 :
vecteur gradient

$$\nabla J(x) = \begin{bmatrix} \frac{\partial J}{\partial x_1}(x) \\ \frac{\partial J}{\partial x_2}(x) \\ \vdots \\ \frac{\partial J}{\partial x_n}(x) \end{bmatrix}$$

Dérivée d'ordre 2 :
matrice hessienne

$$H_J(x) = \begin{bmatrix} \frac{\partial^2 J}{\partial {x_1}^2}(x) & \frac{\partial^2 J}{\partial x_1 \partial x_2}(x) & \dots & \frac{\partial^2 J}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 J}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 J}{\partial {x_2}^2}(x) & \dots & \frac{\partial^2 J}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial^2 J}{\partial x_n \partial x_1}(x) & \frac{\partial^2 J}{\partial x_n \partial x_2}(x) & \dots & \frac{\partial^2 J}{\partial {x_n}^2}(x) \end{bmatrix}$$

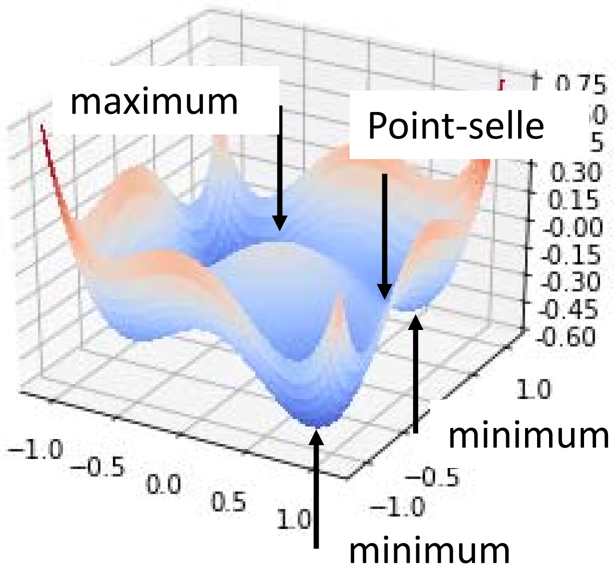
II.2.a Condition de premier ordre

→ Condition d'extremum sur le gradient :

- Dans le cas où J est continument dérivable autour du minimum, alors : $\boxed{\vec{\nabla} J(x^*) = \vec{0}}$
- Il s'agit d'une condition nécessaire, mais non suffisante
- Un point x^* tel que $\vec{\nabla} J(x^*) = \vec{0}$ est appelé « point critique »

→ Un point critique peut être :

- Un minimum (local ou global)
- Un maximum (local ou global)
- Un point-selle (maximum dans une direction, minimum dans une autre)



II.2.a Condition de premier ordre

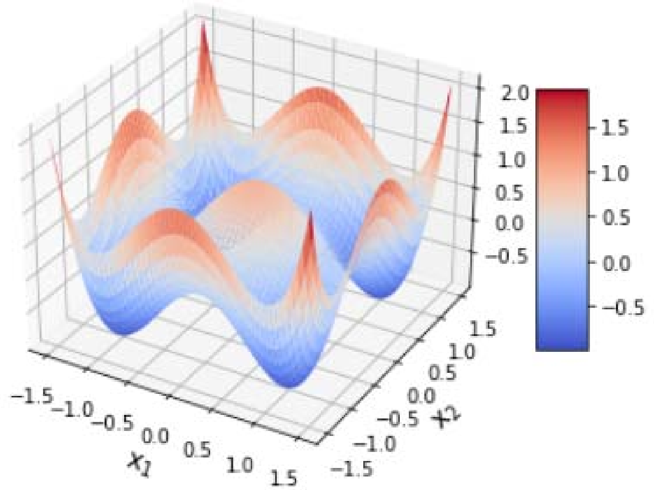
→ Exemple :

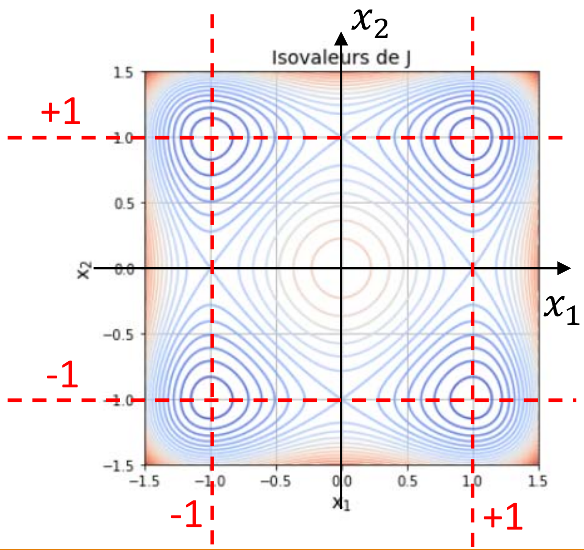
- $J(x_1, x_2) = {x_1}^4 + {x_2}^4 - 2{x_1}^2 - 2{x_2}^2 + 1$

- $\nabla J(x_1, x_2) = \begin{pmatrix} 4{x_1}^3 - 4x_1 \\ 4{x_2}^3 - 4x_2 \end{pmatrix} = 4\begin{pmatrix} x_1(x_1-1)(x_1+1) \\ x_2(x_2-1)(x_2+1) \end{pmatrix}$

- $\nabla J(x_1, x_2) = 0 \implies \begin{cases} x_1 = 0 \;\; ou \; x_1 = -1 \; ou \;\; x_1 = +1 \\ x_2 = 0 \;\; ou \; x_2 = -1 \; ou \;\; x_2 = +1 \end{cases}$

- Bilan : 9 points critiques. Nature ?

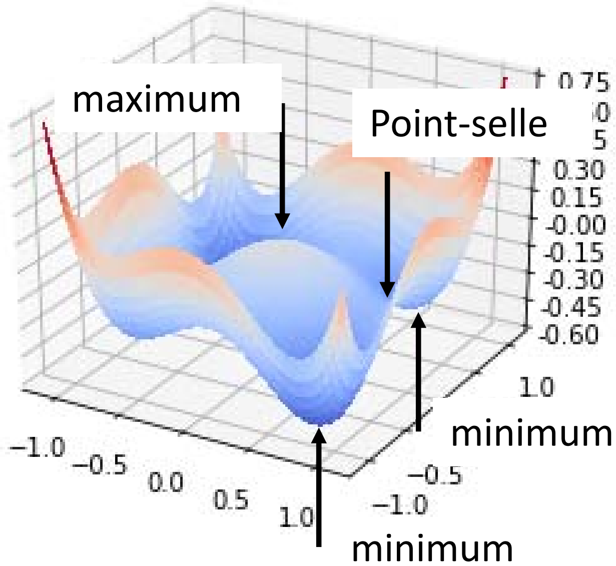




II.2.b Condition de deuxième ordre

→ L'analyse de la matrice hessienne permet de déterminer la nature des points critiques:

- Soient $(\lambda_i)_{i=1,n}$ les valeurs propres de $H_J(x^*)$ (v.p.)
- Si toutes les v.p. sont >0, alors x^* est un minimum local
- Si toutes les v.p. sont <0, alors x^* est un maximum local
- Si certaines v.p. sont >0 et d'autres <0, alors x^* est un point-selle
- S'il existe des v.p. nulles, on ne peut pas conclure



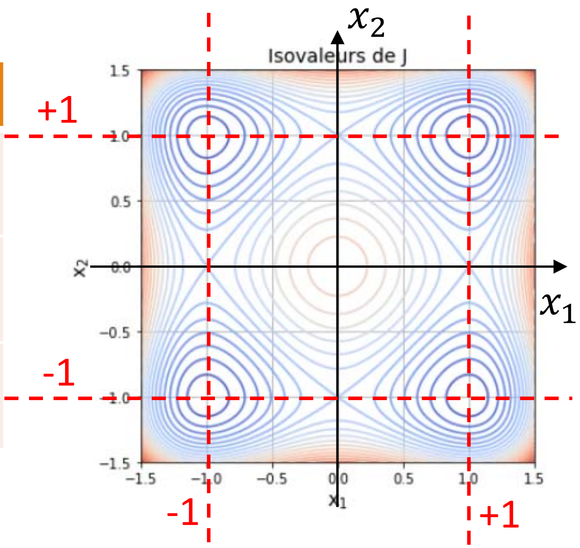
II.2.b Exemple, suite

→ $J(x_1, x_2) = {x_1}^4 + {x_2}^4 - 2{x_1}^2 - 2{x_2}^2 + 1$

→ Matrice hessienne :

- $H_J(x) = 4\begin{bmatrix} 3{x_1}^2 - 1 & 0 \\ 0 & 3{y_1}^2 - 1 \end{bmatrix}$ diagonale, donc les v.p λ_1 et λ_2 sont directement les éléments de la diagonale

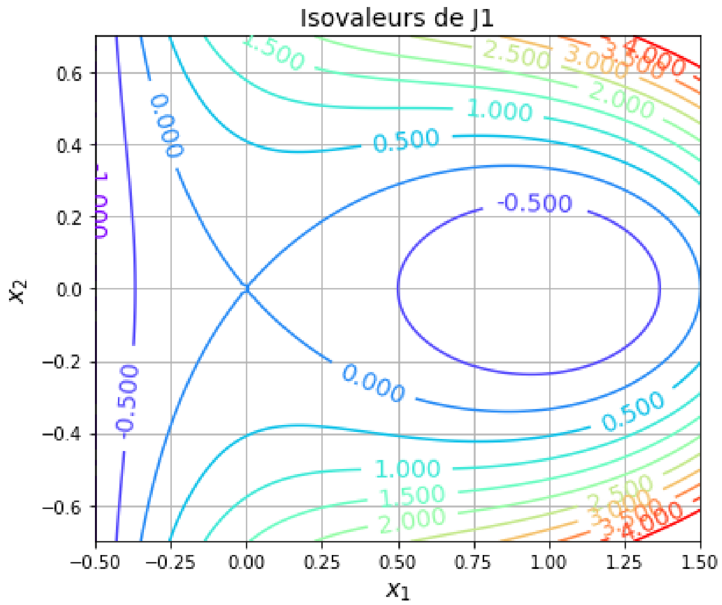
	$x_1 = -1$	$x_1 = 0$	$x_1 = 1$
$x_2 = -1$	$\lambda_1 = 8, \lambda_2 = 8$ minimum	$\lambda_1 = -4, \lambda_2 = 8$ point-selle	$\lambda_1 = 8, \lambda_2 = 8$ minimum
$x_2 = 0$	$\lambda_1 = 8, \lambda_2 = -4$ point-selle	$\lambda_1 = -4, \lambda_2 = -4$ maximum	$\lambda_1 = 8, \lambda_2 = -4$ point-selle
$x_2 = 1$	$\lambda_1 = 8, \lambda_2 = 8$ minimum	$\lambda_1 = -4, \lambda_2 = 8$ point-selle	$\lambda_1 = 8, \lambda_2 = 8$ minimum



II.2.c A vous de jouer

→ soit la fonction $J(x_1, x_2) = 2x_1^3 - 3x_1^2 + 6\,x_1 x_2^2 + 3x_2^2$

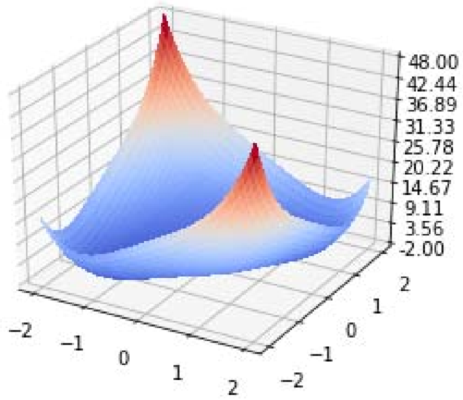
→ D'après les isovaleurs ci-dessous, combien y a-t-il de points critiques ? Faire le calcul.



Quelques résultats d'existence

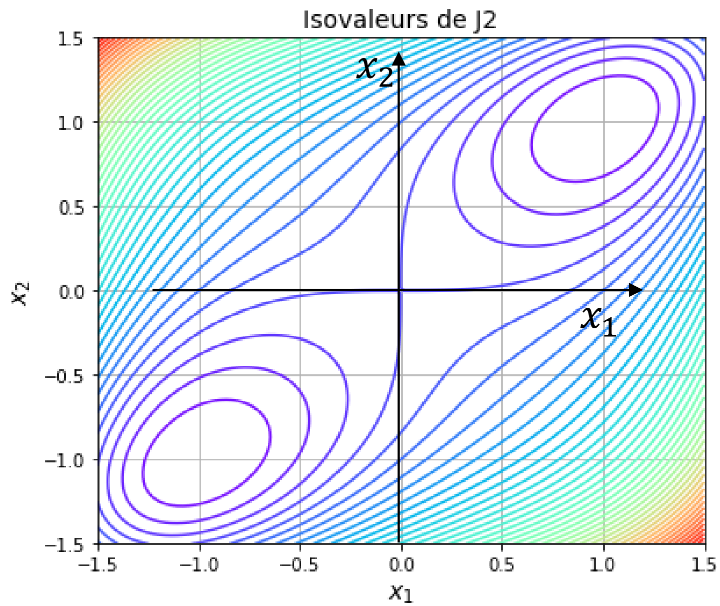
→ Si J est une fonction continue « coercive » :

- Définition : J est coercive ssi : $\lim\limits_{\|x\|\to\infty} J(x) = \infty$
- Alors il existe au moins 1 minimum global (et éventuellement des minima locaux)



→ Exemple :

- $J(x_1, x_2) = {x_1}^4 - 4\,x_1 x_2 + {x_2}^4$
- J est continue et coercive, donc il existe au moins un minimum global
- D'après les isovaleurs ci-contre, combien de points critiques doit-on trouver ?
- Faire le calcul.



Fin du cours, pour aujourd'hui

→ Bilan :

- Généralités sur l'optimisation
- Rappels sur les points critiques d'une fonction de plusieurs variables
- Information sur le déroulement des TP !!! Ca démarre dès la semaine prochaine.

→ A faire, sans tarder :

- Installer au plus vite Python et Jupyter sur votre ordinateur (distribution anaconda)
- Refaire les exercices du cours
- Commencer à jouer avec les Notebook, se mettre à python – les exemples du cours sont disponibles sous moodle
- Commencer le TP1 AVANT la séance